

1

AD-A256 611



AFIT/DS/ENS/92-1

DTIC
ELECTE
OCT 27 1992
S C D

Identification of the Initial Transient in
Discrete-Event Simulation Output Using the
Kalman Filter

DISSERTATION

Mark Austin Gallagher
Captain, USAF

AFIT/DS/ENS/92-1

Approved for public release; distribution unlimited

92-28132



012225

295 P42

AFIT/DS/ENS/92-1

Identification of the
Initial Transient in
Discrete-Event Simulation Output
Using the
Kalman Filter

DISSERTATION

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University

In Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Mark Austin Gallagher, B.S., M.S.
Captain, USAF

December, 1992

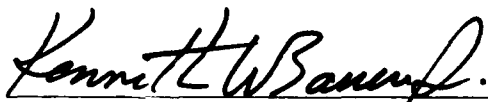
Author	Mark Austin Gallagher
Title	Identification of the Initial Transient in Discrete-Event Simulation Output Using the Kalman Filter
Department	School of Engineering
University	Air Force Institute of Technology
Date	December 1992
Approved for Release	
Distribution	
Final	
A-1	

Approved for public release; distribution unlimited

Identification of the
Initial Transient in
Discrete-Event Simulation Output
Using
the Kalman Filter

Mark Austin Gallagher, B.S., M.S.
Captain, USAF

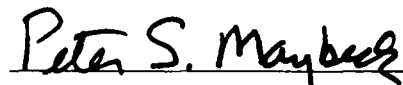
Approved:



Major Kenneth W. Bauer, Jr., Chairman

13 AUG 92

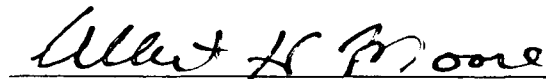
Date



Dr. Peter S. Maybeck, Committee Member

13 Aug 92

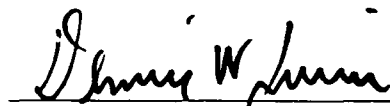
Date



Dr. Albert H. Moore, Committee Member

13 Aug 92

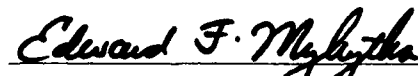
Date



Dr. Dennis W. Quinn, Committee Member

13 Aug 92

Date



Dr. Edward F. Mykyta, Dean's Representative

13 Aug 92

Date

Accepted:



Dr. J. S. Przemieniecki, Senior Dean

4 Sep 92

Date

Acknowledgments

My success in accomplishing this endeavor is the result of the assistance and support of many people. My research committee has been ideal in their guidance and their examples of professionalism. Major Kenneth W. Bauer, Jr., the chairman, is a true mentor. Dr. Peter S. Maybeck is a paragon to me. Dr. Albert H. Moore, Dr. Dennis W. Quinn and Dr. Edward F. Mykytka provided technical insights into different aspects of this research and, more important, valuable inspiration throughout the process. Two additional educators at AFIT deserve my sincere respect and gratitude: Colonel Schuppe, Head of the Operational Science Department, for continually maintaining air-superiority over the evil red forces, and Dr. Bridgman, Dean of Research, for his steady stream of motivational, although satirical, comments.

Dr. John Charnes, after meeting me only briefly at a conference, sent me a copy of his recently completed dissertation. His work often inspired me through my program and was technically very useful to me. Norma Hayes and Jackie Logan, the department secretaries, provided help with many administrative tasks.

Several students have been extremely helpful. Two Master's students plowed very rough ground for me. Chuck Porter was the first to apply Kalman filters to simulation output, and Randy Howard first implemented Multiple Model Adaptive Estimation (MMAE) to simulation output. I appreciate my fellow PhD students, Jean Steppe, Krista Johnson, Dennis Benson, Dan Zeluski, Dustin Johnson, and Jack Taylor, who each at various times commiserated with my academic experiences.

The grant from the Air Force Office of Scientific Research (AFOSR) enabled me to purchase a Sun Sparc2 station, which greatly improved the quality of my work. Thanks to Russel Milliron, my friend and fellow scouter, for his Sun wizardry and more than a few beers.

Besides the contribution of those here at AFIT, my family provided a constant stream of support. Dad and Mom taught me the strength, determination, and courage to pursue my goals. They encouraged me in difficult times and celebrate with me in success. My siblings, Mike, Nancy, Daniel, Paul, John, Ted, and Edward, added their support with phone calls, letters, and prayers.

Besides support, I was extremely fortunate to have the love of my companion, my partner, and my best friend, Nancy Victoria Austin Gallagher, my wife. Her assistance has ranged from computer maintenance and technical editing to pride at my triumphs. I love you, Nancy!

In these three years, I learned a little math, more about myself, and to trust God.

Mark Austin Gallagher

Table of Contents

	Page
Acknowledgments	iii
Table of Contents	v
List of Figures	x
List of Tables	xii
Abstract	xviii
Notation	xix
I. Introduction	1
1.1 Discrete-Event Simulation	2
1.2 Output Analysis	5
1.2.1 Finite-Horizon Simulation Analysis.	6
1.2.2 Infinite-Horizon Simulation Analysis.	7
1.3 Kalman Filters	9
1.4 Research Objective	9
1.5 Preview	10
II. Steady-State Output Analysis Techniques	12
2.1 Introduction	12
2.2 Steady-State Assumptions and Characteristics	14
2.2.1 Covariance Stationarity.	15
2.2.2 Ergodicity.	16
2.3 Output Analysis Issues	17

	Page
2.3.1 Transient Issue.	17
2.3.2 Correlation Issue.	20
2.4 Steady-State Identification	20
2.4.1 Truncation Point Evaluation Criteria.	21
2.4.2 Truncation Approaches.	30
2.4.3 Discrete-Event Simulation of an $M/M/1$ Queue.	37
2.4.4 Requirements for a Good Truncation Point Selection Method.	41
2.5 Summary	41
III. Kalman Filters and Model Estimation	44
3.1 Introduction	44
3.2 Stochastic Processes	45
3.2.1 Linear Stochastic Differential Equations.	45
3.2.2 Dynamics Equation.	48
3.2.3 Measurement Model.	50
3.3 Linear Gaussian Time-Invariant Kalman Filter Algorithm	51
3.3.1 Propagation Stage.	51
3.3.2 Measurement Update Stage.	52
3.3.3 Kalman Filter One Dimensional Example.	55
3.3.4 Constant Kalman Filter Gain.	58
3.3.5 Kalman Filter Optimality.	59
3.4 System Identification	61
3.4.1 Akaike's Estimation Methodology.	63
3.4.2 Multiple Model Adaptive Estimation (MMAE).	68
3.4.3 Transfer Function Technique.	73
3.4.4 <i>A Priori</i> Model Formulation.	74
3.5 Summary	74

	Page
IV. Single-Run Steady-State Identification: Residual Monitoring Approach	75
4.1 Introduction	75
4.2 Determination of the System Structure	76
4.2.1 <i>A Priori</i> Model Formulation.	76
4.2.2 The Kalman Filter Equations.	80
4.3 System Parameter Estimation	83
4.3.1 Identifiability.	84
4.3.2 Estimation Techniques.	85
4.3.3 Variance Estimates.	90
4.3.4 Application of Estimation Procedure.	96
4.4 Truncation Point Selection	102
4.5 Residual Monitoring Truncation-Point Selection Algorithm Steps . .	111
4.6 Monte Carlo Results	113
4.6.1 Autoregressive Test Cases.	115
4.6.2 Autoregressive-Moving Average Test Cases.	145
4.6.3 Analysis.	152
4.7 Summary	162
V. Single-Run Steady-State Identification: Multiple Model Adaptive Estimation (MMAE) Approach	163
5.1 Introduction	163
5.2 Parameter Estimation	164
5.3 Multiple Model Adaptive Estimation (MMAE) Implementation . .	170
5.3.1 Number and Spacing of Filters.	171
5.3.2 Kalman Filter Gain.	173
5.3.3 Filter Probabilities.	181
5.3.4 Declaring the Transient: Heuristic Rules.	181
5.4 Monte Carlo Results	182

	Page
5.4.1 AR(2) with Measurement Noise Data.	182
5.4.2 $M/M/1$ Queue Waiting Time Data.	188
5.5 Summary	197
VI. Multiple Replications Steady-State Identification: Multiple Model Adaptive Es- timation (MMAE) Approach	198
6.1 Introduction	198
6.2 Methodology	198
6.3 Measures of Effectiveness	200
6.4 Monte Carlo Results	201
6.4.1 Results with $\hat{k} = 1.0$	203
6.4.2 Results with Estimated \hat{k}	208
6.4.3 The Effect of Simulation Run Length.	225
6.4.4 The Effect of Filter Spacing.	231
6.5 Summary	233
VII. Multivariate Steady-State Identification	235
7.1 Introduction	235
7.2 VAR(1) with Measurement Noise Model	235
7.3 Parameter Estimation	236
7.4 The Multivariate Truncation-Point Selection Algorithm	240
7.5 Monte Carlo Results	241
7.6 Summary	261
VIII. Summary and Recommendations	263
8.1 Summary	263
8.2 Recommendations for Improvements to the MMAE Truncation-Point Selection Algorithms	264

	Page
8.3 Potential Extensions to Other Discrete-Event Simulation Analysis	
Applications	265
8.3.1 Multivariate Estimation of the Mean Vector.	265
8.3.2 Covariance Matrix Estimation.	266
8.3.3 Improved Control Variates Applications.	267
8.3.4 Enhanced Simulation Run-Length Control.	267
Bibliography	268
Vita	275

List of Figures

Figure	Page
1. Discrete-Event Simulation Model	3
2. Sample Sequence of $M/M/1$ Queue Waiting Times with $\rho = 0.95$ ($E[W_q] = 19$)	38
3. Average of 30 Replications of $M/M/1$ Queue Waiting Times ($\rho = 0.95$)	39
4. Welch's Technique with Moving Average (Window = 500)	40
5. Discrete Sample of White Noise	47
6. Discrete Samples of Brownian Motion	47
7. Propagation Through Time	56
8. Measurement Update	57
9. Steady-State Kalman Filter	60
10. Simulation Output	110
11. Cusum Test Statistic $S_H(j)$	111
12. Types of Transients	116
13. Sample AR(2) with Measurement Noise Data Sequence	174
14. MMAE Filter Probabilities with $\hat{k} = 1.0$	175
15. Residuals of Mean Filter with $\hat{k} = 1.0$	176
16. $H\hat{x}^-(t_n)$ of Mean Filter with $\hat{k} = 1.0$	177
17. MMAE $\hat{\mu}_y(t_n)$ with $\hat{k} = 1.0$	178
18. MMAE $\hat{\sigma}_{\hat{\mu}_y}(t_n)$ with $\hat{k} = 1.0$	179
19. MMAE Filter Probabilities with $\hat{k} = 0.5$	179
20. MMAE $\hat{\mu}_y(t_n)$ with $\hat{k} = 0.5$	180
21. Average Estimated Covariance Matrix $\hat{\Sigma}_{yy} (\hat{\sigma}_{\hat{\Sigma}_{yy,c}})$	243
22. Average Estimated Autocovariance Matrix at One Lag $\hat{\Gamma}_{yy}(1) (\hat{\sigma}_{\hat{\Gamma}_{yy,c}(1)})$. .	243
23. Average Estimated Autocovariance Matrix at Two Lags $\hat{\Gamma}_{yy}(2) (\hat{\sigma}_{\hat{\Gamma}_{yy,c}(2)})$. .	244
24. Average State Transition Matrix $\Phi (\hat{\sigma}_{\Phi_{r,c}})$	244

Figure		Page
25.	Average Kalman Filter Gain \mathbf{K} ($\hat{\sigma}_{\mathbf{K}_{r,c}}$)	244
26.	Average Estimated Covariance Matrix $\hat{\Sigma}_{yy}$ ($\hat{\sigma}_{\hat{\Sigma}_{yr,yc}}$)	250
27.	Average Estimated Autocovariance Matrix at One Lag $\hat{\Gamma}_{yy}(1)$ ($\hat{\sigma}_{\hat{\Gamma}_{yr,yc}(1)}$) . . .	250
28.	Average Estimated Autocovariance Matrix at Two Lags $\hat{\Gamma}_{yy}(2)$ ($\hat{\sigma}_{\hat{\Gamma}_{yr,yc}(2)}$) . . .	251
29.	Average State Transition Matrix Φ ($\hat{\sigma}_{\Phi_{r,c}}$)	251
30.	Average Kalman Filter Gain \mathbf{K} ($\hat{\sigma}_{\mathbf{K}_{r,c}}$)	251
31.	Average Estimated Covariance Matrix $\hat{\Sigma}_{yy}$ ($\hat{\sigma}_{\hat{\Sigma}_{yr,yc}}$)	254
32.	Average Estimated Autocovariance Matrix at One Lag $\hat{\Gamma}_{yy}(1)$ ($\hat{\sigma}_{\hat{\Gamma}_{yr,yc}(1)}$) . . .	254
33.	Average Estimated Autocovariance Matrix at Two Lags $\hat{\Gamma}_{yy}(2)$ ($\hat{\sigma}_{\hat{\Gamma}_{yr,yc}(2)}$) . . .	254
34.	Average State Transition Matrix Φ ($\hat{\sigma}_{\Phi_{r,c}}$)	255
35.	Average Kalman Filter Gain \mathbf{K} ($\hat{\sigma}_{\mathbf{K}_{r,c}}$)	255
36.	Average Estimated Covariance Matrix $\hat{\Sigma}_{yy}$ ($\hat{\sigma}_{\hat{\Sigma}_{yr,yc}}$)	258
37.	Average Estimated Autocovariance Matrix at One Lag $\hat{\Gamma}_{yy}(1)$ ($\hat{\sigma}_{\hat{\Gamma}_{yr,yc}(1)}$) . . .	258
38.	Average Estimated Autocovariance Matrix at Two Lags $\hat{\Gamma}_{yy}(2)$ ($\hat{\sigma}_{\hat{\Gamma}_{yr,yc}(2)}$) . . .	259
39.	Average State Transition Matrix Φ ($\hat{\sigma}_{\Phi_{r,c}}$)	259
40.	Average Kalman Filter Gain \mathbf{K} ($\hat{\sigma}_{\mathbf{K}_{r,c}}$)	259

List of Tables

Table	Page
1. Notation Conventions	xx
2. English Variables and Definitions	xxi
3. Greek Variables and Definitions	xxii
4. Linear System Variable Definitions	xxiii
5. Steady-State Variances and Gain	59
6. Estimation Stopping Criteria	98
7. Truncation Point Selection Criteria	103
8. Cusum Average Run Length (ARL) for $\Delta = 1$	107
9. Cusum Batch Size N_c , Effect on Average Run Length (ARL)	108
10. AR(2) Parameters for Generated Data	115
11. Kalman Filter Model Estimation Summary	118
12. AR(1) with Measurement Noise Model Parameter Estimation Summary . . .	119
13. AR(1) with Measurement Noise Model Variances Estimation Summary . . .	119
14. AR(2) with Measurement Noise Model Parameter Estimation Summary . . .	120
15. AR(2) with Measurement Noise Model Variance Estimation Summary	120
16. AR(1) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$) . .	123
17. AR(1) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$) . .	124
18. AR(1) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$) . .	125
19. AR(1) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$) . .	126
20. AR(2) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$) . .	127
21. AR(2) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$) . .	128
22. AR(2) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$) . .	129
23. AR(2) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$) . .	130
24. AR(1) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	132

Table	Page
25. AR(1) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	133
26. AR(1) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	134
27. AR(1) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	135
28. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	136
29. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	137
30. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	138
31. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	139
32. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model	141
33. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model	142
34. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model	143
35. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model	144
36. ARMA(1,1) Parameters for Generated Data	146
37. Kalman Filter Model Estimation Summary	146
38. AR(2) with Measurement Noise Model Parameter Estimation Summary . . .	147
39. AR(2) with Measurement Noise Model Variance Estimation Summary . . .	147
40. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$)	148
41. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$)	149
42. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$)	150
43. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$)	151
44. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	153
45. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	154
46. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	155
47. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations . . .	156

Table	Page
48. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model	157
49. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model	158
50. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model	159
51. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model	160
52. Correlation Technique Parameter Estimation Summary	167
53. Variance Estimation Summary	168
54. Sum of Squared Residuals for Various Scalar \hat{k}	168
55. Correlation Technique Parameter Estimation Summary	182
56. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$) ($\text{Prob}(\hat{\mu}_y(t_n)) > 0.9$)	183
57. AR(2) Truncation Point \hat{n}_o Evaluations ($\text{Prob}(\hat{\mu}_y(t_n)) > 0.9$)	184
58. Fishman's Number and Size of Batches ($\text{Prob}(\hat{\mu}_y(t_n)) > 0.9$)	186
59. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$) ($\text{ABS}(\hat{\mu}_y(t_n) - \bar{y}) < 1$)	187
60. AR(2) Truncation Point \hat{n}_o Evaluations ($\text{ABS}(\hat{\mu}_y - \bar{y}) < 1$)	189
61. Correlation Technique Parameter Estimation Summary	190
62. Truncation Points \hat{n}_o	191
63. Truncation Point \hat{n}_o Evaluations	191
64. Fishman's Number and Size of Batches	191
65. Average Periodic Output Values for 2,000 Replications ($\mu_y = 10.0$)	193
66. Periodic Truncation Points (Fishman's Number of Batches)	194
67. Truncation Points \hat{n}_o (10 Batches)	195
68. Truncation Point \hat{n}_o Evaluations (10 Batches)	195
69. Periodic Truncation Points (Number of Batches Fixed at 10)	196
70. Theoretical Steady-State Means	203
71. Correlation Technique Parameter Estimation Summary ($\hat{k} = 1.0$)	204

Table	Page
72. Noise Variances Estimation Summary ($\hat{k} = 1.0$)	205
73. Truncation Points \hat{n}_o ($\hat{k} = 1.0$)	206
74. Truncation Point \hat{n}_o Evaluations ($\hat{k} = 1.0$)	207
75. Correlation Technique Parameter Estimation Summary (Estimated \hat{k})	209
76. Noise Variances Estimation Summary (Estimated \hat{k})	210
77. Truncation Points \hat{n}_o (Estimated \hat{k})	210
78. Truncation Point \hat{n}_o Evaluations (Estimated \hat{k})	211
79. Periodic Output Values Points for $M/M/1$ with $\rho = 0.8$ ($\mu_y = 3.2$)	212
80. Periodic Truncation Points for $M/M/1$ with $\rho = 0.8$ ($\mu_y = 3.2$)	213
81. Periodic Output Values Points for $M/M/1$ with $\rho = 0.9$ ($\mu_y = 8.1$)	214
82. Periodic Truncation Points for $M/M/1$ with $\rho = 0.9$ ($\mu_y = 8.1$)	215
83. Periodic Output Values Points for $M/M/1$ with $\rho = 0.95$ ($\mu_y = 18.05$)	216
84. Periodic Truncation Points for $M/M/1$ with $\rho = 0.95$ ($\mu_y = 18.05$)	216
85. Periodic Output Values Points for $M/M/1$ LIFO ($\mu_y = 3.2$)	217
86. Periodic Truncation Points for $M/M/1$ LIFO ($\mu_y = 3.2$)	218
87. Periodic Output Values Points for $M/M/1$ with $L_0 = 10$ ($\mu_y = 3.2$)	219
88. Periodic Truncation Points for $M/M/1$ with $L_0 = 10$ ($\mu_y = 3.2$)	219
89. Periodic Output Values Points for $M/M/1$ L_q ($\mu_y = 3.2$)	221
90. Periodic Truncation Points for $M/M/1$ L_q ($\mu_y = 3.2$)	221
91. Periodic Output Values Points for $E_4/M/1$ ($\mu_y = 1.814$)	222
92. Periodic Truncation Points for $E_4/M/1$ ($\mu_y = 1.814$)	222
93. Periodic Output Values Points for $M/M/2$ ($\mu_y = 2.844$)	223
94. Periodic Truncation Points for $M/M/2$ ($\mu_y = 2.844$)	223
95. Periodic Output Values Points for $M/M/4$ ($\mu_y = 2.386$)	224
96. Periodic Truncation Points for $M/M/4$ ($\mu_y = 2.386$)	224
97. Periodic Output Values Points for $M/M/1/M/M/1M/M/1$ ($\mu_y = 10.233$) . .	226
98. Periodic Truncation Points for $M/M/1/M/M/1M/M/1$ ($\mu_y = 10.233$)	226

Table	Page
99. Periodic Output Values Points for Time-Sharing Computer Model ($\mu_y = 21.38$)	227
100. Periodic Truncation Points for Time-Sharing Computer Model ($\mu_y = 21.38$)	227
101. Periodic Output Values Points for Central Computer Model ($\mu_y = 10.0$)	228
102. Periodic Truncation Points for Central Computer Model ($\mu_y = 10.0$)	228
103. Autocorrelation and Coverage Relationship	229
104. $M/M/1$ $\rho = 0.95$ Parameter Estimation Summary ($\mu_y = 18.05$)	230
105. Truncation Points \hat{n}_o	230
106. Truncation Point \hat{n}_o Evaluations	231
107. $M/M/4$ Truncation Points \hat{n}_o	232
108. $M/M/4$ Truncation Point \hat{n}_o Evaluations	232
109. Open Model Queue Lengths Parameter Estimation Summary	243
110. Truncation Points \hat{n}_o	245
111. Truncation Point \hat{n}_o Evaluations	245
112. Periodic Output Values Points for Open Model (Empty and Idle)	246
113. Periodic Output Values Points for Open Model ($L_{q1}(t_0) = 26$)	247
114. Periodic Truncation Points for Open Model (Empty and Idle)	248
115. Periodic Truncation Points for Open Model ($L_{q1}(t_0) = 26$)	249
116. Open Model (4 Servers) Queue Lengths Parameter Estimation Summary	250
117. Truncation Points \hat{n}_o	251
118. Truncation Point \hat{n}_o Evaluations	252
119. Periodic Output Values Points for the Closed Model (4 Servers)	252
120. Periodic Truncation Points for the Closed Model (4 Servers)	253
121. Closed Model (5 Servers) Parameter Estimation Summary	254
122. Truncation Points \hat{n}_o	255
123. Truncation Point \hat{n}_o Evaluations	256
124. Periodic Output Values Points for Closed Model (5 Servers)	256
125. Periodic Truncation Points for Closed Model (5 Servers)	257

Table	Page
126. Open Model Waiting Times Parameter Estimation Summary	258
127. Truncation Points \hat{n}_o	259
128. Truncation Point \hat{n}_o Evaluations	260
129. Periodic Output Values Points for Open Model Waiting Times	260
130. Periodic Truncation Points for Open Model Waiting Times	261

Abstract

The objective of this research is to improve the available techniques for discrete-event simulation output analysis. Discrete-event simulations are dynamic simulations in which the system states change instantaneously at the occurrence of specified events. Often the distributions of the model outputs attain constant or steady-state characteristics after passing through an initial transient period. The presence of this initial transient can bias estimates of the parameters which characterize the steady-state distributions. One common method to prevent bias from initial data is to delete the data previous to a selected truncation point. Currently, only graphical and heuristic algorithms are available to determine the appropriate initial data truncation point. This research investigates the use of the Kalman filter to select the truncation point for both univariate and multivariate simulation output.

Although residual monitoring is tested, the best truncation-point selection results are obtained with Multiple Model Adaptive Estimation (MMAE). In applying the MMAE technique, the unknown parameter space is discretized and a bank of Kalman filters with one filter associated with each discrete point in parameter space is initialized. The probability of the parameters in each filter being correct is calculated based upon the filter residuals. Weighting the filters' estimates by their probabilities results in a time-varying MMAE parameter estimate. In this application, the unknown parameter is the system mean. The truncation point is selected as the point when the time-varying MMAE mean estimate is within a small tolerance of an assumed state-state mean estimate. The procedure is applied to single output sequences and to the average of multiple replications with excellent results.

Identification of the
Initial Transient in
Discrete-Event Simulation Output
Using the
Kalman Filter

I. Introduction

The objective of this research is to improve steady-state multivariate output analysis techniques for discrete-event simulation through application of Kalman filters. The primary focus is to determine analytically truncation points that reduce the bias from initial conditions in univariate and multivariate discrete-event simulation output sequences for steady-state analysis. This introduction provides an overview of discrete-event simulations, steady-state output analysis, and Kalman filters. Based on this foundation, the research objectives are outlined. The background and details of current output analysis techniques are presented in Chapter II. Chapter III discusses Kalman filters and system identification. In Chapters IV and V, new methodologies for selecting truncation points for single output sequences are proposed and tested. Although the residual monitoring approach applied in Chapter IV is not as effective as the techniques developed in later chapters, the model formulation developed is applied throughout this research. In Chapter VI, the truncation-point selection is applied to situations with multiple replications, and the methodology is extended to multivariate output sequences in Chapter VII. Summary and recommendations are presented in Chapter VIII.

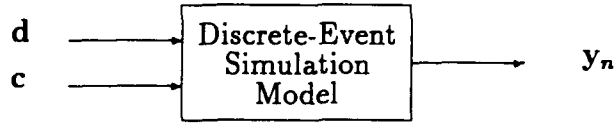
1.1 Discrete-Event Simulation

When no analytical technique is known to investigate a complex system of interest fully, computer simulation often is used. In general, the system under study is defined to be a collection of entities, such as people or machines, that act and interact together toward the accomplishment of some logical end [53:3]. The mathematical or logical relationships, which constitute a model, describe the behavior and interactions between the entities. While the model is an abstraction of the real or proposed system, the model is designed to represent or simulate those characteristics of interest in the real system sufficiently to provide useful insight. In a computer simulation, numerical techniques are used to conduct experiments that estimate the desired characteristics of the model. Since the model abstracts the real world with respect to specific measures of effectiveness, useful inferences about the real system may be made from the simulation model's output.

Computer simulations can be classified as deterministic or stochastic, static or dynamic, and continuous and/or discrete. For deterministic simulations, the exogenous input variables are known. In contrast, stochastic simulations have some deterministic input parameters, but they are driven by random (actually pseudorandom) inputs. For example, a simulation of the partial-differential wave equation would be a deterministic simulation since the inputs are the known boundary conditions. On the other hand, the simulation of a lottery would be a stochastic simulation because the draw would be modeled by a random variate.

Stochastic simulations can again be categorized two ways; static or dynamic. The simulation system state is defined as a collection of variables that describe the system at a particular time relative to the objectives of the study [53:3]. In static simulations, the state does not change with time. Examples of static stochastic simulations are distribution sampling and classical Monte Carlo methods for estimating integrals [103]. On the other hand, the state variables of a dynamic simulation change over time. An example of a dynamic simulation is a single-server queue in which one of the simulation state variables signifies whether the server is busy or idle. This simulation state variable will change depending upon the entities' interarrival and service times.

Figure 1. Discrete-Event Simulation Model



A final way to classify computer simulations is continuous, discrete, or a combination of continuous and discrete. The state variables in continuous simulations usually are described by rate or differential equations, and these values are simulated by approximating the equation solutions over small intervals of time. In contrast, the variables of interest in discrete systems change instantaneously at distinct points in simulated time. These transition points correspond to the occurrences of particular events. For example, in a single-server queue, the server state of busy/idle can only switch either when an entity arrives or when an entity completes service. Combined models using both continuous and discrete state variables exist.

A discrete-event simulation, the topic of this research, is computer simulation which is stochastic, dynamic, and discrete. In Figure 1, a general depiction is shown. The vector \mathbf{d} represents the deterministic input parameters while \mathbf{c} represents the stream of concomitant or control random variates that are generated and used within the simulation model. The model represents the entities and their interactions, and \mathbf{y}_n represents the vector of outputs at time index t_n . The parameters of interest are the values which characterize the distribution of the sequence of output vectors $\{\mathbf{y}_n\}$

Queuing theory is a class of models which study dynamic systems in which entities arrive in a random manner at a service facility [79:304-364]. Since queuing theory provides analytical solutions for classes of systems, these systems are useful for simulation research. Queuing models are classified by the distribution of interarrival times, the distribution of service times, and the number of servers. Using M to designate the exponential (Markovian)

distribution [35:597], the waiting line for a single server with exponential interarrival times and service times is called an $M/M/1$ queue. While this system has an analytical solution, this system also can be modeled by means of a discrete-event simulation. In this example, the control variates c are the streams of exponentially distributed interarrival times and service times. The deterministic inputs d would be the means of the exponential distributions. The model consists of the first-in-first-out (FIFO) service order, which means that the customer who has been waiting the longest gets the server next. More complex queues may have additional deterministic inputs, such as maximum queue length, and the model may also be more elaborate with rules, such as balking or a different order of service. Balking means that under specified conditions, customers would leave rather than enter the queue and wait for service. An example of another service order is last-in-first-out (LIFO), which occurs when newly arriving orders are placed on a stack and next order processed is taken from the top of the stack. The model is a discrete-event simulation since the simulation states, server busy/idle and number in the queue, only change at either an entity arrival or service completion. In this case, a scalar output sequence $\{y_n\}$ might be composed of the waiting time for each entity. Another possible output measure could be the queue length recorded at periodic intervals. Queuing systems often are used in discrete-event simulation research since queuing theory provides an analytical steady-state solution for comparison.

Discrete-event simulation has a wide variety of applications. In particular, discrete-event simulation is used extensively to evaluate systems while they are being designed or to compare various operational alternatives without perturbing the real system. In fact, it is reported as one of the most widely used operations research tools [35:8]. The Air Force, like other organizations with complex problems to analyze, also uses discrete-event simulation. For example, the Air Mobility Command uses the Mobility Analysis Support System, a large discrete-event simulation model, to assess the impact of proposed policies and to evaluate the capabilities of alternative force structures [78]. In addition, the Defense Science Board has recently recommended that more extensive simulations be used to isolate, identify, and quantify potential problem areas during weapon system development and procurement [29].

In 1990, the Department of Defense identified "Simulation and Modeling" as one of the top twenty critical technologies to ensure long-term qualitative superiority of United States weapon systems [74].

Summary articles on discrete-event simulation include Shannon [92], Law [50], Wilson [103], Kelton and Law [45]. Numerous books have been published on discrete-event simulations including Shannon [93], Fishman [23], Kleijnen [47], Banks and Carson [10], and Law and Kelton [53]. In addition, several simulation languages, such as SLAM [77], are used throughout the analysis community.

1.2 Output Analysis

One of the principal objectives of discrete-event simulation is to estimate the characteristics of the model outputs of interest. Since the simulation outputs are functions of the input random variables, the outputs are a sample from a discrete-time stochastic process. The objective of output analysis is to estimate the parameters of the output distribution. Typical parameters of interest are the mean, the variance, a specified quantile, or the probability an output variate is within a specific interval [103].

Typically the output sequence is correlated in time; hence, the estimation process is similar to time series analysis where inferences are made about a stochastic process based on partial observation [12:1]. However, unlike time series analysis, additional realizations can be obtained from separate runs of the model. Since the inputs for discrete-event simulations are typically streams of independent, identically distributed (iid) random variables that are different for each simulation run, each simulation run or replication results in an independent realization of the stochastic process. The goal of output analysis is to make inferences about the stochastic process and the associated distributions based on realizations of the simulation output. However, achieving this goal may be limited by the cost of computer time required to generate the necessary amount of simulation output data [53:523-524].

Since the output sequences are realizations of a stochastic process, any estimates that are derived from them are random variables. Welch [101] summarizes the output analysis

process as follows:

Hence the experimenter must generate from the simulation not only an estimate $\hat{\mu}$ but also enough information about the probability distribution of $\hat{\mu}$ so that he can be reasonably sure that it is close enough to the unknown quantity μ for his purposes. [101:279]

Usually, this is accomplished by using an approximately unbiased estimator, so that $E[\hat{\mu}_y] \approx \mu_y$, and estimating the variance $\sigma_{\hat{\mu}_y}^2$ of the estimator $\hat{\mu}_y$ from sets of sample sequences [101:279]. The estimated variance $\hat{\sigma}_{\hat{\mu}_y}^2$ along with an assumed sampling distribution are used to construct a confidence interval around the estimated mean $\hat{\mu}_y$.

Although a model is an abstraction of reality, in practice, the validity of a model is assumed in output analysis. Therefore, statistical evaluations of alternatives are made based on the model output without considering unmodeled effects. The impact of this assumption is mitigated by the practice of analysts and decision makers to disregard small statistically significant differences as not being practical discriminators between alternatives. Because of the abstraction of reality involved in simulation modeling, discrete-event simulation is better suited for relative comparisons between alternatives than for absolute determination of system performance [15].

Depending upon the objectives of the study, two types of analysis, finite-horizon and infinite-horizon, are typically conducted with discrete-event simulation models. Although this research is limited to infinite-horizon analysis, both types will be described here for completeness.

1.2.1 Finite-Horizon Simulation Analysis. If the parameters of interest are for the output from an initial condition until a natural conclusion, finite-horizon analysis is conducted. The initial condition should be representative of expected conditions in the actual system. For example, in a retail simulation, the initial conditions could correspond to the number of customers and servers in the system when the simulation begins. A reasonable

number of customers may be zero if the simulation is beginning at the store's morning opening. The natural conclusion for a terminating simulation may be the closing of a business, completion of a contract, completion of a military engagement, or end of the planning horizon. Since the period of interest is from the initial condition to the natural conclusion, the model is run through this simulated time. The simulation is repeated to improve the parameter estimates and determine their variability.

1.2.2 Infinite-Horizon Simulation Analysis. If steady-state stationary or cyclic output distributions are of interest [53:528-531], infinite-horizon simulation analysis techniques are used. When steady-state stationary output distribution exist, the output will attain a stationary probability distribution as the length of the simulation run increases. In contrast, steady-state cyclic output will follow a time-varying probability distribution as the simulation output length increases. Although simulation models typically do not have an analytical equation to describe their output, these classifications are similar to the solutions of linear differential equations with time-invariant and time-varying coefficients. This research focuses on estimating the output parameters of discrete-event simulation models that attain a stationary steady-state probability distribution.

While the object of finite-horizon analysis is to estimate the output distribution parameters for specified conditions, infinite-horizon analysis estimates the long-run values of output distribution parameters. In infinite-horizon analysis, particular attention is paid to eliminating the effects of the model's start-up conditions. For example, a logistics system could be modeled to determine the effect of more repair facilities on the expected repair time. Even though the repair facilities may close every day, with the assumption that work picks up where it left off, the sequence of repair times would be unaffected by the breaks. Infinite-horizon analysis indicates the long run capacity of the logistic system regardless of how many failed components are initially in the model.

In infinite-horizon analysis, two problems typically are present. First, the duration of significant effect of the initial conditions on the output is not known a priori. Second, classical

statistical techniques are usually based on independent observations, but the sequence of observations from the simulation model typically are correlated [101:289]. A considerable amount of research in this field has been devoted to develop output analysis techniques which address these two problems.

If the simulation output sequence has a stationary steady-state distribution, then this sequence typically passes through a transient phase, which depends upon the initial conditions. Thereafter, in the steady-state phase, the sequence has an essentially unchanging distribution which is independent of the initial conditions. Rather than the transient phase ending at a particular point, the behavior of the sequence gradually may converge to steady-state behavior [101:277-278]. This convergent behavior depends upon the particular initial conditions and the process dynamics. Since the transient output characteristics are not representative of the steady-state characteristics, including the simulation output from the beginning of a run may bias steady-state estimates. However, determining the point at which the output sequence effectively has attained steady-state is difficult.

While most classical statistic techniques apply to independent replications, simulation output usually is correlated positively within a model run. This correlation reduces the new information available for subsequent output values. Therefore, the run length must be longer to achieve an equivalent level of confidence in parameter estimates. Various techniques exist to attain approximate independence or directly account for the effects of the correlation in the output.

In the following quote, Kelton and Law summarize the current limitations in determining the simulation run length and truncation point so that the sample average of the truncated sequence, \bar{X}_{ml} in their notation, is a good estimate of the mean of the output distribution μ .

What is really needed, then, is a generally reliable method for identifying deletion amounts l , as well as (total) replication length m such that μ_{ml} is sufficiently close to μ to allow us safely to use the \bar{X}_{ml} as being unbiased estimators for μ in the context of a given inferential goal, e.g. c.i. [confidence interval] formulation. [45]

Further background on discrete-event simulation output analysis is in Chapter II.

1.3 Kalman Filters

The Kalman filter is an algorithm to calculate optimal estimates of the state of a stochastic process based on imperfect observations of the system output. Another definition is that the "Kalman filter is simply an *optimal recursive data processing algorithm*" [57:4]. In this definition, optimality can be established by various criteria, such as minimizing any symmetric cost function. The term recursive relates to the two-step nature of the computer program to propagate an estimate and then to update the estimate with an imprecise measurement, repeatedly. The details and theoretical presentation of the Kalman filter are given in Chapter III.

While the Kalman filter is used primarily on engineering applications, statisticians have implemented the algorithm as an optimal Bayesian estimator [67]. In applying the Kalman filter to simulation output, additional information about the stochastic process results in the form of a state-space representation, state estimates, the state estimate's covariances, residuals, and computed residual covariances. This research exploits this additional information to improve simulation output analysis techniques.

1.4 Research Objective

With the increase in computer capabilities and improved simulation languages, larger and more complex simulation models are being built and used. Simulations of complex systems may require large amounts of computer time [53:523]. Generally, these models produce an array of outputs, complicating the necessary multivariate output analysis. For example, in steady-state simulations, the initial transient data should be eliminated to prevent biasing the estimates. However, in order to determine the effective end of the transient phase, each sequence of output variables should be checked for stabilization. The determination of the multivariate estimates along with their associated confidence region and the determination of sufficient simulation run length are also more difficult with larger complex models. Im-

proving the available techniques to conduct univariate and multivariate output analysis of discrete-event steady-state simulation is the general objective of this research.

This research effort focuses on identifying the effective end of the transient phase, or equivalently, the effective beginning of steady-state phase. The objective of this dissertation is to develop improved analytical techniques for determining truncation points that effectively reduce the remaining bias due to the initial conditions. Techniques are proposed for both univariate and multivariate output sequences. The truncation point that are selected by the procedures should be far enough into the simulation run to reduce the bias induced by the transient effects, but not so far that data is deleted unnecessarily, increasing the variance of the estimates.

1.5 Preview

The remainder of this dissertation is organized as follows: Chapter II provides simulation output analysis background. Kalman filters and estimation techniques are described in Chapter III. In Chapter IV, a truncation identification point based on running a single Kalman filter in conjunction with residual monitoring is developed and tested. While the residual monitoring approach did not provide expected results, the same model formulation, developed in Section 4.2, is used throughout this dissertation. Chapter V presents a different approach based on Multiple Model Adaptive Estimation (MMAE) with a bank of Kalman filters. Many popular discrete-event simulation confidence interval techniques are based on a single long run of output [13, 39, 52, 69, 53:551-563]. Because of the run length used with these techniques, the parameter estimates may be biased an insignificant amount by the transient. While the MMAE approach works on a single long run, the value of the additional effort to truncate data which induces a very small bias may be questionable. In contrast, the multiple replication confidence interval technique uses several output sequences. Since each sequence contains initial transient data, the determination of an appropriate truncation point is more important to prevent biased parameter estimates. The MMAE approach is applied to the case of multiple replications of univariate sequences in Chapter VI, and the

method is extended to multivariate output in Chapter VII. The dissertation is completed with conclusions and recommendations in Chapter VIII.

II. Steady-State Output Analysis Techniques

2.1 Introduction

Simulation output analysis can be thought of as a collection of analytical techniques used to make inferences about the probability distributions of the outputs from discrete-event simulation models. These analysis techniques are used to make estimates of parameters which reflect the properties of underlying probability distributions of outputs. Typical parameters that are estimated are the mean, variance, quantiles, or the probability an output observation is within an interval [101, 103].

Parameters for several types of output are typically of interest. Conway [15] classified the types of output as based on either temporary or permanent entities. An output based on a "temporary entity" is associated with a particular entity that flows through the system. For example, the waiting times associated with customers in a queue are temporary variables because each waiting time is associated with a specific customer. The name temporary is used since the variable only exists while the entity is in the system. When the entity completes service and is terminated, the temporary variable no longer exists. Pritsker [77:36] calls the output based on temporary entities "statistics based upon observations". On the other hand, "permanent entities" remain throughout the simulation run. Examples of permanent variables are the number in the queue and busy/idle status of a server. Permanent variables change during the simulation run, and usually the variable's average value over time is the parameter of interest for each such variable. Pritsker [77:36] refers to output which measures permanent variables as "statistics based on time-persistent variables". This research deals with sequences of both "statistics based upon observations" and "statistics based upon time-persistent variables".

The output may be produced at evenly-spaced or unevenly-spaced time intervals. Evenly-spaced outputs are typically simulation state variables that are measured at set intervals of simulated time. For example, the number of entities in the queue could be output

at a predetermined rate. Unevenly-spaced output are generated at varying time intervals. For example, let the output sequence be the waiting times for entities processing through a system. The waiting time associated with each entity is recorded when the entity completes service. Since the entities complete service at random intervals, their waiting times are observed as unevenly-spaced output. While many of the techniques are developed for evenly-spaced output, they can be applied to unevenly-spaced data using an index. For the entities in the simulation of a $M/M/1$ queue, their completion sequence number can be the necessary index. This technique of using indices to represent time is a common practice in time series analysis, and Fishman [20] applies it to simulation output data.

Charnes [12:8] classifies multivariate discrete-event simulation output as synchronous or asynchronous. Synchronous output results when the entire multivariate output vector is produced at the same time. In contrast, asynchronous output results when only a portion of the output vector is obtained from each simulation observation. For example, consider a simulation of a medical clinic that has three types of patients and assume the vector of waiting times by patient type is of interest. Since each patient can only be one of the three types and each patient's completion produces a simulation output, asynchronous output results. Seila [90, 91] proposed two techniques for estimating confidence intervals for asynchronous output. Although this research is limited to synchronous output, a recommendation for further work addresses asynchronous output.

Because of the randomness of the input variates from run to run, "a simulation is a computer-based statistical sampling experiment [50]." Unlike most experimenters, the simulator has significantly more control over the inputs. Therefore, data collection through simulation is very controlled experimentation.

Since the output is a realization or sample of a stochastic process, most output analysis techniques are based on statistical theory. For example, numerous techniques have been developed and are in use to construct a confidence interval for the estimate of the output mean [13, 39, 52, 69, 53:551-563]. Some of the techniques assist in a specific portion of the output analysis task. As an example, the techniques for initial data truncation [24, 44,

104, 105, 101:289-294] are designed to remove unrepresentative data at the beginning of the simulation run to improve the steady-state estimates calculated with other techniques.

In finite-horizon simulation analysis, a simulation model is run from a specified set of initial conditions to a natural ending. For example, a simulation of a ground war may begin with all of the tanks operational and continue until one side is victorious. Running the model repeatedly with different independent random numbers results in independent replications. Since the summary statistics are independent between simulation runs or replications, classical statistical techniques are applied.

In contrast to finite-horizon simulation analysis, infinite-horizon or steady-state analysis techniques are used to estimate the long run or equilibrium characteristics of the system. Some discrete-event simulation models follow a cyclic pattern in steady-state, but this research addresses only simulation models with output that attains a stationary steady-state probability distribution. In most real systems the characteristics of the system change over time, hence the stochastic process may not have a steady-state distribution. In many cases, the changes are slow enough to allow "quasi-static" modeling, and useful inferences can be derived from a stationary steady-state simulation model. In addition, for discrete-event simulation models, the system characteristics are generally held constant throughout the simulation, and therefore, the model output sequences often attain a stationary steady-state distribution [53:530].

The next section presents steady-state discrete-event simulation assumptions and characteristics. The following section reviews the literature on steady-state identification techniques.

2.2 Steady-State Assumptions and Characteristics

If a simulation model output converges to a stationary steady-state distribution, then the probability distribution of the n th observation y_n would approach a limiting distribution for any given initial conditions as n increases, i.e., $\lim_{n \rightarrow \infty} F_{y_n}(\xi | i) \rightarrow F_y(\xi)$ where the i represents initial conditions [53:525]. Since it is difficult, if not impossible, to pick initial

conditions which are representative of the steady-state conditions, the simulation model will pass through a transient phase. Theoretically, the steady state or asymptotic distribution is approached and never actually attained. The goal of steady-state analysis is to run the simulation long enough that the initial conditions no longer have a significant effect on the estimated measures of performance [53:531]. After effectively completing the transient period at n_0 , the output sequence $\{y_n\}$ for $n > n_0$ constitutes a discrete realization of a stochastic process in which each y_n follows the steady-state distribution. Although members of this sequence are not independent, they often approximately constitute a covariance-stationarity process [53, 101]. In addition, many of the output analysis techniques implicitly assume ergodicity. The next two sections describe the concepts of covariance stationary and ergodicity.

2.2.1 Covariance Stationarity. Covariance stationarity requires that the mean of the output sequence μ_y is constant and the elements of the covariance matrix are finite and constant through time, and the degree of correlation between random variables in the sequence depends only on the distance between the variables and not on their absolute location [53, 57, 77]. In particular for a scalar output sequence $\{y_n\}$,

$$\mu_y \equiv E[y(t_n)]$$

$$\sigma_y^2 \equiv E[(y(t_n) - \mu_y)^2]$$

For times s and t ,

$$\gamma_y(t - s) \equiv \text{Cov}[y(t), y(s)] = E[(y(t) - \mu_y)(y(s) - \mu_y)]$$

$$\gamma_y(0) = \sigma_y^2$$

$$\gamma_y(i) = \gamma_y(-i)$$

$$\rho_y(i) = \gamma_y(i)/\gamma_y(0)$$

where $\gamma_v(i)$ is the autocovariance and $\rho(i)$ is the autocorrelation at a lag of i . Covariance stationarity implies that

$$E[y(t_n)] = \mu_y, \text{ a constant for all } n$$

$$E[y(t_n)^2] < \infty \text{ and a constant for all } n$$

$$\gamma_v(t-s) = \gamma_v(r-q) \text{ if } |t-s| = |r-q|$$

For multivariate output,

$$\mu_y \equiv E[y(t_n)]$$

$$\Sigma_{yy} \equiv E[(y(t_n) - \mu_y)(y(t_n) - \mu_y)^T]$$

and the autocovariance matrix Γ_{yy} is defined

$$\Gamma_{yy}(t-s) \equiv E[(y(t_t) - \mu_y)(y(t_s) - \mu_y)^T]$$

$$\Gamma_y(0) = \Sigma_{yy}$$

and covariance stationarity implies that

$$E[y(t_n)] = \mu_y \text{ is a constant vector}$$

$$E[y(t_n)y^T(t_n)] \text{ is a finite and constant matrix}$$

$$\Gamma_{yy}(t-s) = \Gamma_{yy}(r-q) \text{ if } |t-s| = |r-q|$$

2.2.2 Ergodicity. Another important concept is ergodicity, which Maybeck [57] defines as follows:

A process is *ergodic* if any statistic calculated by averaging over all members of the ensemble of samples at a fixed time can be calculated equivalently by time-

averaging over any single representative member of the ensemble, except possibly a single member out of a set of probability zero. [57:144]

The advantage of a realization of an ergodic process is that the properties of the random variable can be estimated from a single realization. Parzen [75:73] proves that the sequence of sample means $\{\hat{\mu}_y(n), n = 1, 2, \dots, \infty\}$, is ergodic if $\sigma_{\hat{\mu}_y(n)}^2 \rightarrow 0$ as $n \rightarrow \infty$. Parzen [75:74-75] further proves that a necessary and sufficient condition for an ergodic process is that the elements of the autocovariance matrices are bounded and the covariance, between the sample mean $\hat{\mu}_y(n)$ and the last observation y_n , tends to zero as n increases. Parzen [77:74] states that a sufficient condition for a covariance stationary process to be ergodic is that the autocovariance function $\gamma_y(i)$ tends to zero as i increases to infinity. The importance of this result in the simulation application is that an estimator of the mean approximately equals the process mean if the variance of the estimator goes to zero as the run length increases to infinity [77:727]. Many of the discrete-event simulation output analysis techniques, such as all the confidence interval approaches based on a single run, implicitly assume the simulation output is an ergodic sequence.

2.3 Output Analysis Issues

Most of the research on steady-state simulation analysis has dealt with some aspect of the two problems, the initial transient or the correlated output. Many of the techniques can be categorized as either multiple-run techniques or single-run techniques. Multiple runs achieve independence by separate stochastic realizations. However, the initial transient must be deleted from each of the initial runs. The single-run techniques use one long simulation run so that the transient must only be deleted once. However, accounting for the correlation in the output is a critical problem for the single-run techniques. The next two sections detail the transient and correlation problems.

2.3.1 Transient Issue. The initial transient or startup problem is induced by initial conditions which are not representative of the unknown steady-state output distribution.

Conway [15] states that the transient components typically decay geometrically in time and convergence to steady-state conditions can be quite slow. Odoni and Roth [73] numerically solved the Kolmogorov forward differential equations to describe the transient behavior of certain queuing systems. They stated,

The rate at which a queue converges to its steady-state characteristics, independently of the system's initial state, eventually becomes (for large values of time t) dominated by an exponential term of the form $\exp(-t/\tau)$ where τ is a characteristic of the queuing system.

For their queuing systems, the characteristic τ is found to vary with powers of the coefficients of variation of the interarrival and service times, and it is independent of the initial conditions selected. Their finding of exponential decay to the steady-state output distribution agrees with other researchers, such as Conway [15].

Once the output of a simulation effectively has attained its stationary steady-state probability distribution, each observation has the same cumulative density function $F_y(\xi)$, regardless of the initial conditions of the simulation model. While the steady-state probability distribution is attained only in the limit, the point after which steady state is effectively achieved is labeled, using Welch's notation [101:295], as n_0 . Effectively attained means that the error caused by assuming the system is in steady state is negligible [15]. Therefore, \hat{n}_o is an estimate of the end of significant effects of the transient. Thus, $E[y_n] \neq \mu_y$ for $n \leq n_0$ but analysts assume $E[y_n] = \mu_y$ for $n > n_0$. If the mean of the output sequence is calculated using all the data,

$$\hat{\mu}_y = \frac{1}{N} \sum_{n=1}^N y_n$$

then the expected value does not equal the steady-state mean; $E[\hat{\mu}_y] \neq \mu_y$. On the other hand, if the output sequence prior to n_0 is deleted, then the mean of the remaining data would be an unbiased estimator of the steady-state stochastic process mean:

$$E[\hat{\mu}_y] = E \left[\frac{1}{N - n_0} \sum_{n=n_0+1}^N y_n \right] = \frac{1}{N - n_0} \sum_{n=n_0+1}^N E[y_n] = \mu_y$$

For this reason, a current practice is to delete or truncate the beginning portion of the output sequence. The difficulty arises in determining how many of the initial observations to delete. Not deleting enough initial data may bias the resulting steady-state estimates, whereas truncating too much data unnecessarily increases the variance of the steady-state estimates.

While the steady-state values do not depend on the initial conditions, the length and rate of convergence do depend on the initial conditions for discrete-event simulation models [53:551]. Numerous researchers [43, 71, 53:551] have investigated the effects of various initial conditions.

Madansky [56] examines ($M/M/1$) queues with no data deletion. He shows that if the objective is to minimize mean square error (MSE) in the estimate of the number in the system, the mode and not the steady-state mean is the optimal number of queued observations for an initial condition. In addition, increasing simulation run length is more effective than adding replications in reducing the MSE [56]. Extending Madansky results to the case where deletion is allowed is not obvious.

Kelton and Law [46] analyze the transient behavior of different queues. Their findings indicate that, for systems with several servers, starting empty and idle with a build-up to steady-state is much slower than beginning overcongested and drawing-down to steady state. Initial conditions can be specified based on the output of pilot runs. While most of the literature favors the mode of the number in the system as the best initial condition, Kelton and Law [46] prefer the mean or greater.

Other initialization routines have been proposed. Law and Kelton [53] credit Glynn for suggesting a one-time pass through the transient to develop starting conditions. Kelton [43] proposes a random initialization which reduces the severity and duration of the initial transient period.

However, while these papers demonstrate the significant impact of the initial conditions, in actual practice, these techniques are difficult to implement. First, real-world multi-

variate simulations have many queues and factors which would be required to be initialized. In addition, many of these queues are maintained internal to the simulation model, and the corresponding queue statistics are not included in the output. Therefore, the best initial conditions for these queues could not be estimated from the output of previous runs without modifying the model's output routines. Regardless, since setting specific initial conditions shortens, but does not eliminate the transient [15], an improved algorithm to identify the end of significant effects of the transient is worthwhile.

2.3.2 Correlation Issue. Besides dealing with the problem of the initial transient, we must also contend with the problem of correlated output. The effect of this correlation is the difficulty, not only in estimating the stochastic process variance, but also in estimating the variance of estimators. For example, the variance of the sample mean is below:

$$\sigma_{\hat{\mu}_y}^2 = \frac{1}{N^2} \left\{ \sum_{n=1}^N \sigma_y^2 + 2 \sum_{j=1}^{N-1} \sum_{k=j+1}^N \text{Cov}[y_j y_k] \right\}$$

Usually discrete-event simulation output is correlated positively. Ignoring the positive correlation results in underestimating the variance of the estimator $\hat{\mu}_y$ and the corresponding confidence interval being too small.

2.4 Steady-State Identification

Since the run length of any simulation is finite, the true steady-state or asymptotic distribution is never really achieved. However, for the purposes of analysis, steady-state is considered to have begun when the initial conditions no longer have an effective impact on the distribution of the output sequence. This point is labeled as n_0 using Welch's notation [101:295].

The extent of the transient phase is difficult to determine. Law and Kelton [53:527] indicate that the sample mean of the distribution of waiting times for $M/M/1$ queue with traffic intensity ρ of 0.8 is very near the theoretical steady-state level within 500 customers. Eilon and Chowdhury [19] report that for a simulation of an $M/M/1$ queue with high traffic

intensity, a substantial difference between the theoretical and the estimated steady-state mean waiting time may exist for a batch of 1,000 customers, even after 20,000 customers. While Eilon and Chowdhury attribute this to initialization bias, it may also be induced by process variability and high autocorrelation.

Steady state is attained when the cumulative density function of the output sequence $F_y(\xi)$ stabilizes. Welch [101:290] proposes that the most straightforward approach to identify steady state would be to estimate $F_{y_n}(\xi)$ based on y_n for each n , but he admits the computing requirements would be impractical. Theoretically, many of the techniques would work with higher moments, such as convergence of the output sequence variance; however, the increased variance in estimating higher moments makes detection of the stabilization more difficult. Therefore, although stabilization of the mean is a necessary, but not a sufficient condition for steady state, the widely applied techniques are based on mean estimates stabilizing [53:545-551].

Before discussing the techniques for estimating n_0 that have been proposed in the literature, the next section examines evaluation criteria. The following section describes different approaches to select a truncation point to eliminate the biased initial data along with published comparisons. An example of a simulation model of an $M/M/1$ is presented. The last section of the chapter summarizes the requirements for a reliable and useful truncation point selection technique.

2.4.1 Truncation Point Evaluation Criteria. This section reviews criteria that may be applied to evaluate truncation point selection algorithms. The algorithms are generally evaluated in terms of the effectiveness of the truncated sequences to make correct inferences about a known parameter. The earliest criterion considered in the literature is mean squared error (MSE) of the mean estimator for the truncated sequence. Later researchers [44, 52, 104, 105] have recommended using confidence interval widths and coverage rates for evaluating the effectiveness of truncation sequences. Therefore, the confidence interval construction techniques are described in this section. The first confidence interval technique

addressed is the method of replications, which is appropriate when multiple simulation runs are conducted with univariate or multivariate output. Besides the method of replications, confidence interval construction techniques designed for a single simulation run, nonoverlapping batch means (NOBM) and overlapping batch means (OBM), are also described. Since both NOBM and OBM require that a batch size be specified, Fishman's method to select a batch size [22] is included in the discussion. After reviewing the confidence interval techniques that may be used to evaluate the effectiveness of truncation sequences, Schruben's technique to test for initial bias in a truncated sequence [87] is described. The performance measures of MSE, confidence interval coverage and half widths, and the percentage of truncated sequences passing Schruben's initial bias test are described in this section and applied in Monte Carlo analyses in later chapters.

Truncation points are selected to delete data at the beginning of the simulation which is not representative of the steady-state values. Not deleting enough data biases the resulting estimates, whereas deleting too much data increases the variance of the initial parameter estimator [53]. The inverse effect of sample size can be seen through the derivation for the variance of the estimated mean, shown in Equation (1). This derivation assumes a covariance-stationarity process. Therefore, the autocovariance $\gamma_y(t-s) \equiv \text{Cov}[y(t), y(s)]$ is

constant for any set interval between t and s .

$$\begin{aligned}
\sigma_{\hat{\mu}_y}^2 &= E[(\hat{\mu}_y - \mu_y)^2] \\
&= \frac{1}{N^2} E \left[\sum_{i=1}^N (y_i - \mu_y) \sum_{j=1}^N (y_j - \mu_y) \right] \\
&= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E[(y_i - \mu_y)(y_j - \mu_y)] \\
&= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \gamma_y(i - j) \\
&= \frac{1}{N^2} \sum_{h=-N+1}^{N-1} (N - |h|) \gamma_y(h) \\
&= \frac{1}{N} \sum_{h=-N+1}^{N-1} \left(1 - \frac{|h|}{N}\right) \gamma_y(h) \\
&= \frac{\gamma_y(0)}{N} \sum_{h=-N+1}^{N-1} \left(1 - \frac{|h|}{N}\right) \rho_y(h) \\
&= \frac{\sigma_y^2}{N} \sum_{h=-N+1}^{N-1} \left(1 - \frac{|h|}{N}\right) \rho_y(h)
\end{aligned} \tag{1}$$

or equivalently,

$$= \frac{\gamma_y(0) + 2 \sum_{h=1}^{N-1} \left(1 - \frac{h}{N}\right) \gamma_y(h)}{N}$$

Assuming that the autocorrelation decreases sufficiently fast as the lag increases, $\lim_{h \rightarrow \infty} \rho_y(h) = 0$, then for large N ,

$$\sigma_{\hat{\mu}_y}^2 \approx \frac{\sigma_y^2}{N} \left[\sum_{h=-\infty}^{\infty} \rho_y(h) \right] = \frac{1}{N} \left[\sum_{h=-\infty}^{\infty} \gamma_y(h) \right] \tag{2}$$

Since truncating the data reduces the sample size to $N' = N - n_0$, the variance of the mean estimator increases. In other words, while initial data truncation reduces the bias, it also increases the variance of the mean estimate.

In order to account for the tradeoff between bias reduction at the expense of variance induction, numerous researchers [20, 21, 14, 100] used the mean square error (MSE). For a point estimator $\hat{\theta}$ of θ , the variance is $\sigma_{\hat{\theta}}^2 = E[(\theta - \hat{\theta})^2]$ and the bias is $B(\hat{\theta}) = E[\hat{\theta}] - \theta$. Therefore,

$$\begin{aligned}
\text{MSE}(\hat{\theta}) &= E[(\hat{\theta} - \theta)^2] \\
&= \sigma_{\hat{\theta}}^2 + B(\hat{\theta})^2
\end{aligned}$$

Since the MSE of an estimator is the variance plus the squared bias, it appears to be an excellent criterion to determine the correct tradeoff between increased variance and reduced bias.

Fishman [21] analyzes the effectiveness of truncation on data from a first order autoregressive AR(1) model, $(y_n - \mu_y) = \alpha(y_{n-1} - \mu_y) + \epsilon_n$ with $(-1 < \alpha < 1)$. The contribution of the initial condition decreases by α^n as the observation number n increases. Fishman compares no data deletion and a deletion point \hat{n}_0 such that the initial value y_1 contributed less than a predetermined percent to $y_{\hat{n}_0}$. For a range of α 's and a range of truncation points based on various percentages, Fishman shows analytically that deleting the initial transient reduces bias, but may significantly increase variance. In fact, in terms of MSE, Fishman concludes that truncation may not always be advisable. Turnquist and Sussman [100] extend Fishman's work to single server queues with various traffic intensities and found that, to minimize MSE, truncation may not be worthwhile. Wilson and Pritsker [105] report that Blomqvist shows MSE is minimized by no data deletion for a special class of queuing systems. For Markov processes, Cheng [14] makes an argument for one continuous run versus replications based on MSE under the assumptions of no initial truncation and the bias being a monotonically decreasing function of sample size.

These researchers use MSE of the mean estimator of the truncated sequence as a criterion to evaluate truncation point selection algorithms. Later researchers [44, 52, 104, 105] consider other criteria, in particular, confidence interval half widths and their associated coverage rates. The next part of this section describes three confidence interval construction techniques, the method of replications, nonoverlapping batch means (NOBM) and overlapping batch means (OBM), that are applied in the Monte Carlo analyses.

The objective of simulation analysis is to make statistical inferences about the distributions of output sequences. These inferences are made typically in terms of a point estimator and an associated confidence interval for a prescribed level of confidence. In these procedures, a tradeoff exists between the width of the confidence interval and the level of confidence. Since the interval typically is chosen to be symmetric about the point estimator, the half width of the confidence interval generally is used for comparisons between techniques. These same criteria can be used to evaluate the effect of truncating output sequences at different points to eliminate the initial transients.

In Monte Carlo analyses, the percentage of times that confidence intervals contain a known steady-state parameter is called the *coverage rate*. Nominal coverage rates are reflected by the level of confidence, $1 - \alpha$, where α is the theoretical proportion of confidence intervals that do not contain the true parameter. Law and Kelton [53:561-562] report that actual coverage rates have been found to be considerably less than the expected coverage rates. Wilson and Pritsker [104, 105] say the actual coverage attained is a more important criterion than MSE to the objective of simulation output analysis. A bias in the point estimator would cause the confidence intervals to contain the true parameter significantly less often than the nominal level of the specified confidence interval. Therefore, Wilson and Pritsker [104] propose a method to fix the confidence interval width for various truncation points and initial conditions in order to compare the alternative deletion techniques in terms of their actual coverage rates. Kelton and Law [44] state, "by looking at criteria other than point estimator mean-square-error (in particular, true c.i. [confidence interval] coverage probability and expected width), deletion can improve statistical validity with only minor losses in efficiency." Kelton and Law [45] studied the effects of various deletion amounts and number of replications for an AR(1) process, and they found that deletion improves the confidence interval coverage rates to acceptable levels without substantially widening the confidence interval half widths.

Confidence intervals for parameters of discrete-event simulation output distributions can be constructed in a number of ways. If multiple simulation runs are made, the method of replications is used. Since each replication is initialized with a different pseudorandom number generator seed, the sample output sequences are independent. Let $y_m(N)$ be the mean of the m th replication with N sequential observations. Based on M replications, the mean estimate is

$$\hat{\mu}_y = \left(\frac{1}{M}\right) \sum_{m=1}^M y_m(N)$$

with sample variance of

$$\hat{\sigma}_{\hat{\mu}_y}^2 = \left(\frac{1}{M}\right) \sum_{m=1}^M \frac{(y_m(N) - \hat{\mu}_y)^2}{M - 1}$$

A $100(1 - \alpha)$ percent confidence interval is constructed using

$$\hat{\mu}_y \pm t_{d,1-\alpha/2} \hat{\sigma}_{\hat{\mu}_y} \quad (3)$$

where $t_{d,1-\alpha/2}$ is the upper $1 - \alpha/2$ critical value for the Student's t distribution with degrees of freedom $d = M - 1$.

The method of replications can be extended to multivariate output to determine a confidence ellipsoid and its associated volume. For M replications with N observations each with S responses, let $\mathbf{y}_{m,s}(N)$ be the mean response of the m th replication for the s th response. Let $\hat{\mu}_y$ be the vector of average responses across replications, and let the sample covariance matrix $\hat{\Sigma}_{yy} = \frac{1}{M-1} \sum_{m=1}^M (\mathbf{y}_m(N) - \hat{\mu}_y)(\mathbf{y}_m(N) - \hat{\mu}_y)^T$. Under the assumption that the $\mathbf{y}_m(N)$ are distributed normally with mean μ_y and covariance Σ_{yy} , the confidence region, as shown in Rubinstein and Marcus [80], is derived from:

$$\Pr \left\{ M(\hat{\mu}_y - \mu_y)^T \hat{\Sigma}_{yy}^{-1} (\hat{\mu}_y - \mu_y) \leq \frac{(M-1)S}{M-S} F_{S,M-S;1-\alpha} \right\} = 1 - \alpha \quad (4)$$

where $F_{S,M-S;1-\alpha}$ denotes the $(1 - \alpha)$ quantile of the F distribution with S and $M - S$ degrees of freedom. The associated volume of the confidence region is calculated via

$$V = \frac{1}{S} C(S) |\hat{\Sigma}_{yy}|^{\frac{1}{2}} \left[\frac{S(M-1)}{M(M-S)} F_{S,M-S;1-\alpha} \right]^{\frac{S}{2}} \quad (5)$$

where $C(S) = \frac{2\pi^{\frac{S}{2}}}{\Gamma(\frac{S}{2})}$ and Γ is the gamma function.

If only one long univariate simulation run is conducted, two commonly applied confidence interval construction techniques are nonoverlapping batch means (NOBM) and overlapping batch means (OBM). NOBM is described in almost every simulation textbook [23, 51, 53, 77]. NOBM is similar to independent replications except that the replications' means are replaced with batch means. Each of the M batch means $y_m(B)$ is based on B sequential observations. If the batch size B is sufficiently large, the batch mean estimators are

approximately independent [101]. With the batch means replacing the replications' means, the confidence interval is constructed in the exact manner as the method of replications, Equation (3). Meketon and Schmeiser [69] propose that it is the number of batches, not their independence, which is critical. By overlapping the batches in OBM, they state that the asymptotic variance can be reduced to $\frac{2}{3}$ of NOBM variance when using the same batch size. Schmeiser and Song [82] published FORTRAN codes for both NOBM and OBM.

To use either NOBM or OBM, an appropriate batch size must be determined. Fishman [22, 23:238-241] proposes a methodology to determine batch size. Fishman's algorithm is based on the correlation between the batches calculated with

$$C_M = 1 - \frac{\sum_{m=1}^{M-1} (y_m(B) - y_{m+1}(B))^2}{2 \sum_{m=1}^M (y_m(B) - \hat{\mu}_y)^2} \text{ where } \hat{\mu}_y = \frac{1}{M} \sum_{m=1}^M y_m(B) \quad (6)$$

If the batch means are independent and normally-distributed, with as few as eight batches, C_M is approximately normally-distributed with a mean of zero and a variance of $\frac{M-2}{M^2-1}$. Based on C_M , a hypothesis test for statistical independence can be tested. In this research, if C_M is outside the 90 percent confidence interval centered at zero, the null hypothesis of independence is rejected. Fishman's algorithm begins with a batch size of one, $B = 1$ with $M = N$, and doubles the batch size until the null hypothesis is not rejected. If the batch size B increases until there are less than eight batches, $M < 8$, the algorithm fails to find an appropriate batch size.

Since recent researchers recommend confidence interval coverage and half widths, rather than MSE, to evaluate the effectiveness of truncation point selection algorithm, this research employs those criteria. Specifically, the coverage rates and average half widths for the truncated sequences are reported. For univariate single run Monte Carlo analyses, the confidence interval techniques of NOBM and OBM using Fishman's batch size algorithm are applied. For multiple runs Monte Carlo analyses, the method of replications coverage rates and average half widths are reported. In addition, Schruben's initial bias test [87] is used as a criterion for univariate sequences. The next part of this section describes Schruben's test.

For a univariate sequence, Schruben [87] uses the concept of a Brownian bridge [75] to develop a statistical test to determine if initialization bias remains in the truncated output sequence. The development of Schruben's technique starts by describing the output sequence $\{y_n\}$ as the sum of the mean $\mu_y(t_n)$ and a noise term $\eta(t_n)$

$$y(t_n) = \mu_y(t_n) + \eta(t_n) \text{ for } n = 1, 2, \dots, N \quad (7)$$

In steady state, $\mu_y(t_n) = \mu_y$ and $E[\eta(t_n)] = 0$. If initialization bias is present, then $\mu_y(t_n)$ changes in the sequence. Schruben [87] assumes that the noise sequence $\{\eta(t_n)\}$ is stationary and ϕ -mixing. The property ϕ -mixing intuitively means that "distant future behavior of the process is almost independent of the present or past behavior of the process [87]."

Thinking of the output sequence as a signal $\mu_y(t_n)$ plus noise $\eta(t_n)$, Schruben develops a transformation that converts the noise to a Brownian bridge process. Schruben summarizes a Brownian bridge as follows:

The limiting stochastic process used is a standard Brownian bridge $\beta_t; 0 \leq t \leq 1$, i. e., Brownian motion on the unit interval conditioned to start and return to zero. β_t here is the process analogue to the standard normal random variable used in applications of the classical central limit theorem. The Brownian bridge process has continuous sample paths and four notable properties:

1. $\beta_0 = \beta_1 = 0$,
2. $E[\beta_t] = 0; 0 \leq t \leq 1$,
3. $\text{Cov}(\beta_{t_1}, \beta_{t_2}) = \min(t_1, t_2)(1 - \max(t_1, t_2))$, and
4. Sets of β_t have a jointly normal distribution. [87]

In his paper, Schruben shows that under the assumption that the $\eta(t_n)$ are independent identically distributed random variables, the following transformation,

$$B_{k/N} = \frac{k[\frac{1}{N} \sum_{n=1}^N \eta(t_n) - \frac{1}{k} \sum_{n=1}^k \eta(t_n)]}{\sqrt{N}\sigma_\eta} \text{ for } k = 1, 2, \dots, N$$

meets the first three criteria of a Brownian bridge process. The last criterion is argued in the limit.

Under the assumption of no initialization bias, the transformed sequence can be modeled as a Brownian bridge process. However, an initialization bias caused by $\mu_y(t_n) \neq \mu_y$ induces a non-constant signal. This signal causes the transformed sequence to have an unusually large value. Under the assumption of no signal, the transformed sequence of noise is a Brownian bridge process. Since a Brownian bridge process follows a jointly normal distribution, a transformed sequence without noise should follow a normal distribution at each observation. The presence of a non-constant signal, which corresponds to initialization bias, is apparent as a statistical outlier.

The actual test for initialization bias is based on the null hypothesis that no bias is present. This technique finds the maximum value of a transformed sequence,

$$\hat{s} = \max \left\{ \frac{i(\frac{1}{N} \sum_{n=1}^N y_n - \frac{1}{i} \sum_{n=1}^i y_i)}{\sqrt{N}} \right\} \text{ where } i = 1, 2, \dots, N \quad (8)$$

Let \hat{i} be the index of the observation which is the maximum value, \hat{s} . Letting $\hat{t} = \frac{\hat{i}}{N}$, Schruben shows that

$$\hat{x} = \frac{\hat{s}^2}{N\sigma_{\hat{\mu}_y}^2 \hat{t}(1 - \hat{t})} \quad (9)$$

has approximately a χ^2 distribution with 3 degrees of freedom when no initialization bias is present. However, to calculate \hat{x} requires the true variance of the mean estimator $\sigma_{\hat{\mu}_y}^2$. Since the ratio of two approximately independent χ^2 random variables follows approximately an F distribution [36:220], if $\hat{\sigma}_{\hat{\mu}_y}^2$ is estimated with d degrees of freedom,

$$\hat{h} = \frac{\hat{s}^2}{3N\hat{\sigma}_{\hat{\mu}_y}^2 \hat{t}(1 - \hat{t})} \quad (10)$$

has approximately an F distribution with 3 and d degrees of freedom. Schruben applies Fishman's [23:247-255] p th-order autoregressive, $AR(p)$, formulation to estimate the process

variance and associated degrees of freedom. Fishman derives an estimate of $\hat{\sigma}_{\hat{\mu}_y}^2$ as

$$\hat{\sigma}_{\hat{\mu}_y}^2 = \frac{Q_d}{N(1 - \hat{\phi}_1 - \hat{\phi}_2 - \dots - \hat{\phi}_p)^2} \quad (11)$$

where Q_d is an estimate of the variance of the zero-mean normally-distributed noise term of the AR(p) process. Degrees of freedom d for this estimator are

$$d = \frac{n(1 - \hat{\phi}_1 - \hat{\phi}_2 - \dots - \hat{\phi}_p)}{2p - 2 \sum_{s=1}^p (p - 2s) \hat{\phi}_s} \quad (12)$$

Schruben recommends estimating $\hat{\sigma}_{\hat{\mu}_y}^2$ using the last half of the data to prevent a bias from the initial data. An initialization bias caused by $\mu_y(t_n) \neq \mu_y$ induces a non-constant signal and is apparent as a statistical outlier. Based on the same concepts, Schruben, Singh, and Tierney [88] develop more powerful tests based on *a priori* knowledge about the transient behavior.

This completes the literature review on truncation point evaluation criteria. For evaluating the potential for future applications of a truncation technique, a good set of criteria measures the quality of confidence intervals constructed with the truncated sequence. The confidence interval quality is measured by the average confidence interval widths and realized coverage rate. If the initial data bias is removed, the coverage rate should be very near the nominal rate. Excessive deletion is apparent at wide average confidence interval widths. Another potential criterion is the pass rate of Schruben's test for initialization bias detection, a "state-of-the-art" approach.

2.4.2 Truncation Approaches. Numerous techniques to select the deletion or truncation point n_0 are proposed in the literature. This section reviews the techniques based on Gafarian, Ancker and Morisaku's [24] survey paper and Wilson and Pritsker's [104, 105] survey. Both papers were published in 1978. Since these surveys, Kelton and Law [44] and Welch [101] have proposed two truncation point selection techniques. Following the description of these methodologies, Schruben's technique for truncating multivariate sequences is

reviewed.

Gafarian, Ancker, and Morisaku [24] conduct a comprehensive study of the commonly used techniques to select the deletion point. They evaluate the following five techniques:

Conway's Rule Truncate the series until the remaining first value is neither the minimum nor maximum of the remaining data.

Conway's Rule (Modified) Find the first observation that is not the minimum or maximum of the earlier rejected data values.

Crossing-of-the-Mean Rule As proposed by Fishman, a running cumulative mean is maintained. When the number of times the next data value crosses the cumulative mean (switches from being above to below or vice-versa) reaches a specified number, select that as the truncation point.

Cumulative-Mean Rule Using a set of exploratory simulation runs, the grand cumulative mean is plotted. The point of stabilization is selected at the truncation point.

Gordon's Rule Since the variance of the mean estimate should be inversely proportional to the sample size, $\sigma_{\hat{\mu}_v}^2 \propto \frac{1}{N}$, Gordon recommends calculating and plotting $\hat{\sigma}_{\hat{\mu}_v}$ on a log-log graph with increasing increments of N . The N for which the graph begins following a straight line with slope of minus one half is selected as the beginning of steady-state.

The five techniques are compared based on accuracy and precision of selecting the "true" truncation point. The "true" truncation point is determined by using the theoretical expectation of the waiting time for observations proceeding through queues. This "true" truncation point is selected as the observation with a waiting time within five percent of the steady-state value.

After empirically testing the rules with Monte Carlo analysis for a variety of queuing systems, Gafarian, Ancker, and Morisaku [24], state, "none of the five rules is satisfactory and that they should not be recommended for use by practitioners."

Wilson and Pritsker [104, 105] survey the various simulation start-up techniques. Only the two additional techniques in their summary are presented here.

Fishman [20] develops an estimate of n_0 using the comparison between correlated observations and independent observations to achieve an equivalent reduction in the variance of the same mean. The estimate is based on the following result which is derived in Equation (2):

$$\sigma_{\hat{\mu}_y}^2 \approx \frac{\sigma_y^2}{N} \left[\sum_{k=-\infty}^{\infty} \rho_y(k) \right]$$

For independent observations,

$$\rho(k) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases}$$

and

$$\sigma_{\hat{\mu}_y}^2 = \frac{\sigma_y^2}{N_{\text{independent}}}$$

Since an independent observation by definition is independent of the initial conditions, set $N_{\text{independent}} = 1$ and equate the two equations for the variance of the mean estimator:

$$\frac{\sigma_y^2}{1} = \frac{\sigma_y^2}{N} \left[\sum_{k=-\infty}^{\infty} \rho_y(k) \right]$$

Solving for N results in a point in the data sequence independent of the initial conditions. Therefore, the estimated truncation point is $\hat{n}_o = \sum_{k=-\infty}^{\infty} \rho_y(k) - 1$. The difficulty with this procedure is that estimating the correlation coefficients is difficult. For finite-order autoregressive (AR) processes, Fishman [23] expresses the truncation point in terms of the autoregressive parameters.

Wilson and Pritsker [105] state that Schriber [84] proposes the truncation point as the beginning of a set number of batches for which all of the batch means fall within a small interval. Wilson and Pritsker tested these techniques and conclude that the truncation rules of thumb "are very sensitive to parameter misspecification, and their use can result in excessive truncation."

In 1983, Kelton and Law [44] developed a simulation run length determination and truncation point selection approach based on linear regression. The estimated mean for M replications and small batches of size B over the time index n , $\hat{\mu}_y(M, B, n)$, is modeled as the population mean μ_y plus a zero-mean noise term $\eta(t_n)$:

$$\hat{\mu}_y(M, B, n) = \mu_y + \eta(t_n) \quad (13)$$

Under the assumption that $E[y(t_n)]$ converges monotonically to μ_y , their method starts at the end of the data, fitting regressions on blocks of data, until the hypothesis of zero slope is rejected. The size of the block of data is maintained by removing data at the end. Batching is employed to ensure normally-distributed disturbances and is not intended to eliminate autocorrelation in the noise term. Hence, generalized least squares, rather than ordinary least squares, is applied. They desired a conservative approach because, to them, reducing the bias is more important than minimizing the deletion amounts.

In Kelton and Law's algorithm, M simulation replications are each run for a relatively short duration. The averages across replications are calculated, and the sequence of averages is batched. A regression line is fit to the last half of the batch means and is tested for zero slope. If the slope is nonzero, each of the M simulation replications is continued to increase the length of their output sequence. After increasing the run lengths, the procedure is applied again. If the slope tests to be zero, the simulation replication lengths N are fixed at their current lengths. Regression lines are fit to earlier segments of data until a nonzero slope is encountered. The point of the first nonzero slope becomes the truncation point \hat{n}_0 .

Kelton and Law's algorithm presents three difficulties. First, continuing simulation run lengths after stopping the model may be difficult in applications. To restart the replications, the entire simulation state vector must be recorded and the model must be reset to these conditions prior to restarting. In any relatively complex simulation model, this restarting is difficult to implement. In addition, besides the difficulting in restarting the models, the algorithm requires generalized least squares, which allows for correlated data, to be implemented.

Second, the detailed procedure requires the analyst to specify nine parameters. These nine parameters are the number of replications, the initial simulation run lengths, the amount of increment increases to the run lengths, the maximum run length, the number of batches, the minimum and maximum deletion portions, the significance level for the test for zero slope, and the maximum number of segments over which a fit is made. Determining appropriate values for these parameters may prove difficult. Finally, Kelton and Law's algorithm assumes monotonic convergence to the steady-state mean. They recommend further work to identify the algorithm parameters and to extend the method to non-monotonic approaches to steady state. Kelton and Law's Monte Carlo results are used as a base line for comparison in Chapter VI.

While Kelton and Law's method assumes monotonic convergence to a steady-state mean, Odoni and Roth [73] determine initial conditions under which the expected number in the queue does not move monotonically toward the steady-state value. Kelton [42] and Murray and Kelton [71] examine the transient behavior of a large class of queuing models and also note non-monotonic approach to the steady-state means. They also demonstrate that initial conditions greatly affected the time to decay within a specified interval of the steady-state value.

The same year Kelton and Law published their techniques, 1983, Welch [101:290-294] proposed a truncation procedure that is relatively easy to implement. This method involves graphically observing the convergence of the mean estimator. Since for $n > n_o$, $E[y_n] = \mu_n \approx \mu$, Welch recommends plots of the output sequence $\{y_n\}$. To reduce the variance, he recommends averaging the y_n across replication runs. The larger the number of replications, the easier the stabilization can be detected. The decision can be simplified further if the high frequency fluctuations are removed with a moving average. The moving average window size is increased until the stabilization on the plot is obvious to the practitioner. In addition, Welch suggests plotting confidence intervals about the means for various n to give an additional indication about when the transient effects are insignificant.

Law and Kelton [53:552] state that, for Welch's procedure to be statistically correct,

separate sets of runs should be used to identify the transient phase and for parameter estimation. They state that, if the transient phase is short compared to the run length, then it is “probably safe” to use the same set of runs to determine the warm-up period and calculate parameter estimates.

Law and Kelton [53:545-550] further recommend Welch’s [101:290-294] plotting technique with subjective assessment as the “simplest and most general” approach to detect the completion of the transient phase. However, this technique is only appropriate for univariate output sequences. Law and Kelton also state that, in general, one replication is not sufficient to estimate n_o .

To this point, this literature review has concentrated on identifying the effective completion of transient phase for the univariate output case. Schruben [86] proposes a method to select a truncation point for multivariate output with multiple replications.

For a multivariate output sequence, initialization bias may affect some output measures and be insignificant for others. Besides serial correlation within one of the output sequences, cross and lagged-cross correlations should be considered. Schruben [86] demonstrates the limitation of trying to examine each output sequence independently for stabilization. He illustrates a case in which each of the output sequences has stabilized, but the correlation between the sequences has not stabilized.

Schruben’s test is based on Hotelling’s two sample T^2 statistic [28:166-169]. Each output vector \mathbf{y}_n consists of S responses, and M replications are run. Schruben notes that to use Hotelling’s T^2 requires $M > S$, and he further recommends that $M - S > 5$. The procedure begins by dividing the output vectors into N_B small batches with batch size B of 5 to 10. These batches are not constructed to reduce serial correlation but to support the normality assumption of the test statistic. An average vector is calculated within a batch for each replication, $\bar{\mathbf{y}}_{m,i}(B)$, and these are averaged across the replications.

$$\bar{\bar{\mathbf{y}}}_i(MB) = \frac{1}{M} \sum_{m=1}^M \bar{\mathbf{y}}_{m,i}(B) \text{ for } i = 1, 2, \dots, N_B \quad (14)$$

Presuming that the run is long enough for the last batch, $i = N_B$, is at equilibrium, the sample covariance matrix is calculated by

$$S_{N_B}(MB) = \frac{1}{M-1} \sum_{m=1}^M (\bar{y}_{m,N_B}(B) - \bar{y}_{N_B}(MB))(\bar{y}_{m,N_B}(B) - \bar{y}_{N_B}(MB))^T \quad (15)$$

When the number of replications is low, Schruben recommends using a group of the last batches to calculate the sample covariance matrix. The Hotelling's two-sample test statistic T^2 for each batch i is given by

$$T_b^2 = \frac{M^2(M-S)}{(M-1)(2M)S} [\bar{y}_i(MB) - \bar{y}_{N_B}(MB)]^T S(N_B)^{-1} [\bar{y}_i(MB) - \bar{y}_{N_B}(MB)] \quad (16)$$

The truncation point can be selected when T_b^2 shows consistency with an $F_{\alpha,S,M-S}$ distribution [86]. Specifically, beginning with the first batch, batches are discarded until $T_b^2 \leq F_{\alpha,S,M-S}$ for a selected value of α .

Schruben evaluates his technique through a Monte Carlo analysis. He graphs the coverage rates based on the desired confidence versus the observed confidence level for two sets of data. The first data set is the remaining observations after using his criteria, and the second set has the previous "biased" batch added in. With a large number of replications, the effects of bias becomes much more apparent because, as the number of replications increases, the confidence region decreases. For a biased estimator, the confidence region shrinks around the incorrect point. Bias is therefore apparent by the observed coverage rate of the confidence region on the true parameter being considerably lower than the nominal confidence level. Schruben tests his method on response times and utilization from a time-sharing computer model [3], employing 100 replications with a batch size of 3. Using his truncation criterion results in the desired confidence level equaling the observed coverage rate. However, when one batch before the appropriate truncation point is added to the data set, the observed confidence level decreases significantly. The effectiveness of truncation points selected by the technique proposed in Chapter VII are compared with the effectiveness of the truncation points selected by Schruben's algorithm.

2.4.3 Discrete-Event Simulation of an M/M/1 Queue. The M/M/1 queue is selected because its theoretical mean is known and the simulated output is complex and highly variable. Using the notation that the mean of the exponentially-distributed interarrival times is $\frac{1}{\lambda}$ and the mean of exponentially-distributed service times is $\frac{1}{\mu}$, the traffic intensity or server utilization is $\rho = \frac{\lambda}{\mu}$ (unitless).

In the literature, two example output sequences commonly are considered. One output is waiting time in the system for a sequence of entities. This is an example of output that Pritsker [77:36] classifies as “statistics based on observations”. The other sequence is a periodic output of the number in the simulation queue, which is an example of “statistics based on time-persistent variables”.

The expected waiting time $E[W_q]$ in steady state is

$$E[W_q] = \frac{\rho}{\mu - \lambda}$$

with variance

$$\sigma_{W_q}^2 = \frac{\lambda\mu}{(1 - \rho)^2}$$

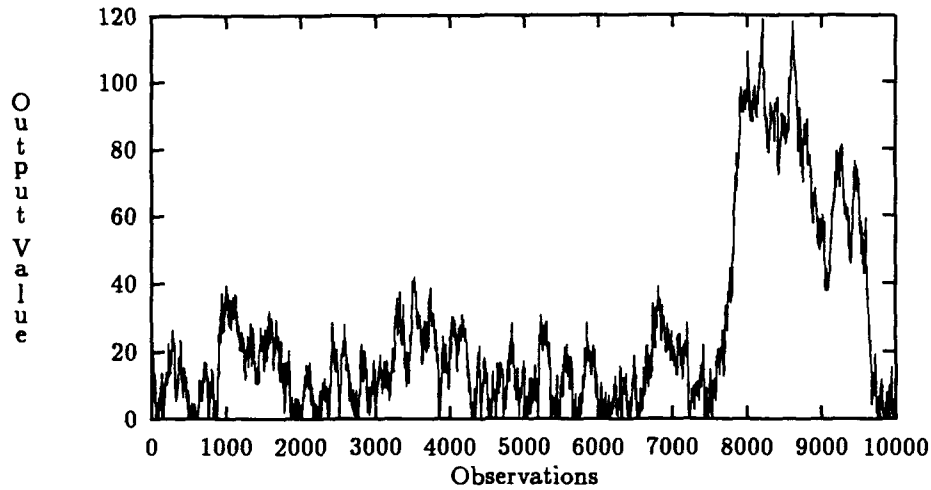
The steady-state expected length of the queue $E[L_q]$ is given by

$$E[L_q] = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda}$$

As traffic intensity ρ increases, so does the correlation in the output. Therefore, as traffic intensity increases, the significant effects of the initial transient should persist longer. The queue becomes unstable and grows without bound when $\rho \geq 1$ since entities are arriving faster than entities are served on the average. Figures 2 through 4 depict simulations where ρ is 0.95 and the initial conditions are an empty queue and an idle server to show a case with high autocorrelation and a long initial transient.

Figure 2 depicts a typical sequence of waiting times for entities processing through an M/M/1 queue with first-in-first-out service order, no balks, and infinite queue capacity. For

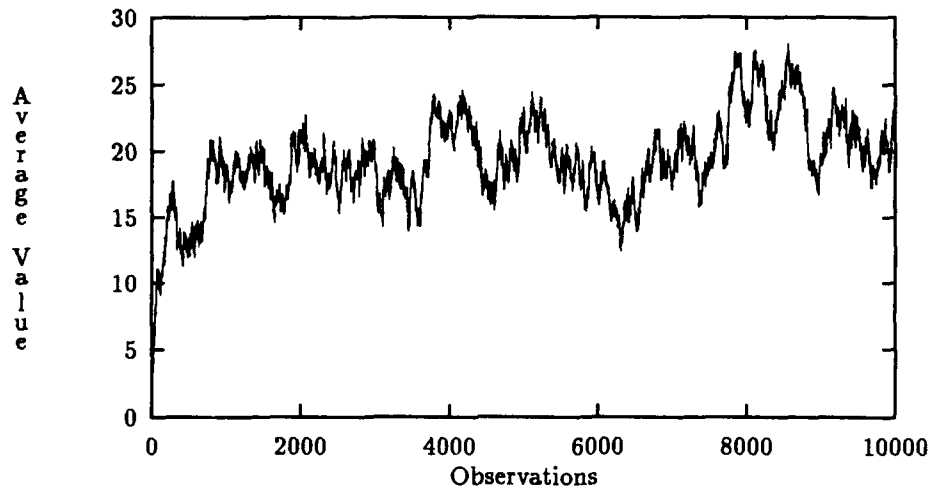
Figure 2. Sample Sequence of $M/M/1$ Queue Waiting Times with $\rho = 0.95$ ($E[W_q] = 19$)



this example, the traffic intensity ρ is 0.95 and the steady-state expected waiting time $E[W_q]$ is 19.0 time units. While the ratio of the arrival and service rates determines the traffic intensity, their magnitudes determine the average waiting time. The congestion and long waits experienced by the 7,500th through 9,500th entities are typical and occur periodically when the system becomes congested. The length and duration of long lines increase with traffic intensity.

The initial condition in this queue is empty and idle, so the first entity through has no waiting time, $W_q(t_0) = 0$. Because of this initial condition, the probability of early entities having long waiting times is significantly less than from the steady-state output distribution for waiting times. For the system to be stable, that is $\rho < 1$, the mean arrival rate must be less than the mean service rate, or equivalently, the mean interarrival time must be greater than the average service time. Thus, in steady-state, the server will occasionally be idle, and waiting times of zero occur. The empty and idle initial conditions cause the transient values to have lower values, but not necessarily to be statistical outliers of the steady-state output distribution. The initial transient values simply have distributions with a sequence of lower

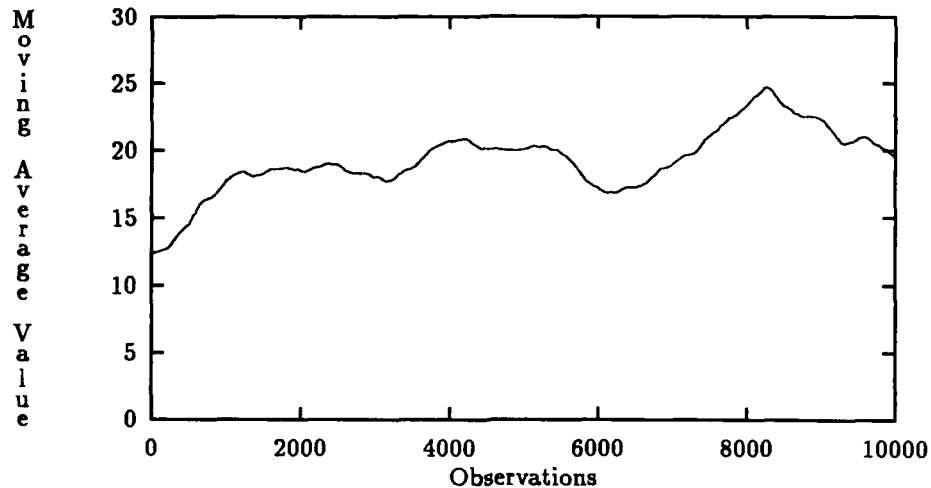
Figure 3. Average of 30 Replications of $M/M/1$ Queue Waiting Times ($\rho = 0.95$)



means.

Averaging over independent replications of the simulation gives an empirical indication of the time-dependent output distribution mean. Figure 3 shows the average of 30 output sequences. The first observation in each sequence is zero, and by about the 1000th observation, the average is very near the steady-state average of 19.

Figure 4. Welch's Technique with Moving Average (Window = 500)



Welch's technique [101], which is described earlier, recommends processing the sequence of averages through a moving average. In the moving average, each observation is replaced with the average of a set number of preceding observations, the same number of following observations, and the original observation. The number of preceding observations is called the moving average window size. At the beginning and end of the sequence, the average is taken over the points that would be in the complete window. Welch's technique assumes that the steady-state distribution is gradually approached so that the output distribution of nearby observations should be very similar. The moving average smooths the sequence of averages across replications. Welch recommends increasing the window size until the transient is apparent. Figure 4 depicts the moving average with a window size of 500 for the average of 30 output sequences. Since the sequence appears to become level for the 1000th to 4000th observation, the initial transient appears effectively to be complete by the 1000th observation.

2.4.4 Requirements for a Good Truncation Point Selection Method. Gafarian, Ancker and Morisaku [24] use five criteria for comparing truncation techniques. Their criteria are accuracy, precision, generality, cost, and simplicity. The first two, accuracy and precision, relate only to situations with predefined "true" truncation points. Confidence interval half widths and coverage rates are generally accepted as the best criteria for measuring the success of a technique. Generality means that "the rule performs well across a broad range of systems and a broad range of parameters within a system." Cost reflects the computer time required for the algorithm. However, since simulation models themselves can require significant amounts of computer time, almost any technique will have negligible impact on the total computer time required. Simplicity ensures that the rule is usable by the average simulation practitioner and implies that the technique should not require too many variables to be set by the user. Past techniques are criticized for being too sensitive to parameter settings and for not providing rules to specify the necessary parameter settings [105, 77:753].

General recommendations are against trying to determine steady state based on cumulative statistics. Their values typically lag behind the current statistics, resulting in excessive deletion of data [15].

2.5 Summary

The typical assumptions and issues for steady-state output analysis of discrete-event simulations are discussed. Common assumptions embedded in most analysis techniques are covariance stationarity and ergodicity of the output sequence. Covariance stationarity in a univariate sequence implies that the output sequence has a constant mean, finite and constant variance, and the covariance for any two observations separated by any set interval is the same. For discrete-event simulation output analysis, ergodicity implies that the estimator of the mean is unbiased if its variance decreases to zero as the number of observations increases to infinity.

Output analysis issues include dealing with the initial transient and correlated output. The initial transient is induced since the initial conditions are not representative of unknown

steady-state conditions. The typical approach of dealing with the transient or warm-up period is initial data deletion or truncation. Deleting too much initial data increases the variance of the steady-state parameter estimates, whereas not deleting enough data may result in biased estimates. Determining the appropriate truncation point is the objective of this research.

Positive correlation in simulation output reduces the information obtained from each additional observation. Since the statistical techniques for independent observations do not apply directly, each output analysis technique must deal with the correlation of the output.

The next major section in this chapter deals with steady-state or initial transient identification. In other words, steady-state identification is determining the point where the sequence should be truncated to eliminate the estimation bias induced by the initial conditions. While the steady-state output distribution is only reached in the asymptotic limit as run length increases, the simulation output passes a point where steady state is effectively attained. This means that the error induced by assuming that steady state has been attained is statistically insignificant.

The section on steady state identification reviews the literature for evaluation criteria. Currently, the most widely accepted criteria are the resulting confidence interval width and realized coverage rates for the truncated sequences. Schruben's [87] initialization bias test can also be used.

The literature review continues with proposed truncation point selection algorithms and published evaluations of these algorithms. A general conclusion of the literature is the lack of a sound analytical approach for the truncation of a univariate output sequence. The chapter concludes with criteria for a good initial-data truncation method.

One of the major sources of difficulty in simulation output analysis is a result of correlation in the output sequence. In both Schruben's [87] and Kelton and Law's [44] approaches, the output sequence is assumed to be representable as a constant steady-state mean plus noise. Since the output sequence is generally correlated, the mean being constant

implies the noise accounts for the correlation. Schruben [87] assumes the noise is ϕ -mixing, but he only demonstrates that the transformation to a Brownian bridge works if the noise sequence is independent. Kelton and Law [44] use generalized least-squares to account for the autocorrelation.

In a different approach to account for correlation in the output sequence, this research applies a filter with an autoregressive model to account for the correlation in the noise. Chapter III presents the linear system theory and Kalman filter equations for this approach, and univariate truncation point selection methodologies based on these concepts are proposed in Chapters IV and V. Although the residual monitoring approach in Chapter IV is not as successful as the MMAE approach in Chapter V, Section 4.2 provides the model formulation used throughout this research. The transient effects are less significant with a single run, so the truncation method is applied to cases with multiple replications in Chapters VI and VII. Chapter VI provides the algorithm and results for univariate output sequences, and in Chapter VII the method is extended for multivariate output sequences. Chapter VIII summarizes this research.

III. Kalman Filters and Model Estimation

3.1 Introduction

Kalman [40] and Kalman and Bucy [41] developed the Kalman filter. Kalman's contribution was to combine the work of the German mathematician Gauss with the state-space vector representation from linear system theory. The state vector is a set of variables such that the present state vector with future inputs completely describe system behavior. This definition of state-space representation is limited to Markov processes since the Markov property implies that some set of sufficient statistics at the current time provides as much information about the system as the complete time history of the process. In a state-space representation, system equations are written to depict the state dynamics and measurement relationships. Kalman's original derivation is based on the fact that the updated state estimate at time t_n , $\hat{\mathbf{x}}(t_n^+)$, is the orthogonal projection of the true state $\mathbf{x}(t_n)$ onto the subspace spanned by the measurement history $\mathbf{Z}(t_n)$ [57:235].

The Kalman filter is an optimal recursive next-step prediction algorithm. After initialization, the discrete Kalman filter continues with a two-step procedure. The first step propagates the best estimate of the state through time. The second step uses the information contained in a noise-corrupted measurement to update the prediction. The algorithm repeats again by propagating this estimate to the next time interval. These steps of propagation and update are repeated for each of the available observations.

The notation employed in this chapter follows engineering control theory conventions, specifically Maybeck [57, 58, 59]. Estimated parameters are indicated with a hat, and vectors are in boldface, such as the state vector estimate $\hat{\mathbf{x}}(t_n)$. At each point in time, two state estimates are used; the propagated estimate, prior to the measurement update, is indicated with a minus, $\hat{\mathbf{x}}(t_n^-)$, and the state estimate after the measurement update is marked with a plus, $\hat{\mathbf{x}}(t_n^+)$. Alternative Kalman filter notation and descriptions include Meinhold and Singpurwalla [68] and Harvey [33].

Simulation output is a discrete realization of a stochastic process. In the next section, the stochastic process is approximated as the solution to a linear stochastic differential equation driven by Brownian motion. The stochastic process is defined in terms of the states that are estimated by the Kalman filter, which may or may not be the simulation states or the simulation observations. Since simulation output calculated with a digital computer is discrete, rather than using the differential equation directly, an equivalent discrete-time difference model [57:170] is applied. The discrete-time Kalman filter is discussed in Section 3.3. The final major section in this chapter is on system identification. While techniques for system identification are reviewed after describing the Kalman filter, the actual model formulation and system identification are not discussed until the next chapter.

3.2 Stochastic Processes

Before showing the Kalman filter formulas in the next section, some underlying theory of stochastic processes is discussed. This section presents a linear stochastic differential equation and the associated dynamics model and measurement model.

3.2.1 Linear Stochastic Differential Equations. Assume the underlying stochastic process can be described or approximated with a linear stochastic differential equation,

$$dx(t) = F(t)x(t)dt + G(t)d\beta(t) \quad (17)$$

where x is a vector of the filter-design system states

F is the dynamics system matrix

G determines which system states are affected by noise

β is the dynamics driving noise

The system is linear if $F(t)$ is not a function of the state values $x(t)$. Further, assume that the dynamics driving noise can be modeled as Brownian motion. Brownian motion has the following characteristics [57:148,155]:

- Brownian motion has independent increments in time, and
- the increments are normally distributed random variables such that for any time instants, t and t' ,

$$E[\beta(t)] = 0$$

$$E[(\beta(t) - \beta(t'))(\beta(t) - \beta(t'))^T] = \int_{t'}^t Q(\tau) d\tau$$

$$\beta(t_0) = 0 \text{ by convention}$$

The matrix $Q(\tau)$ represents the diffusion of the process. The term diffusion is from an early application of Brownian motion for modeling the movement of gas molecules. In other applications, $Q(\tau)$ determines how fast the states in the process are changing.

Heuristically, Brownian motion is the result of white (uncorrelated in time) noise passed through an integrator. In reverse, the hypothetical derivative of Brownian motion is white noise. The derivative is hypothetical because Brownian motion is nondifferentiable. Heuristically, Brownian motion is a continuous process which has corners everywhere [57:152].

A discrete sample of scalar white noise generated by sequencing 100 independent standard normal random variables is shown in Figure 5. The notation $w_d(t_n, \omega_1)$ indicates w for white noise, subscript d for discrete, t_n for time, and ω_1 to indicate which realization from the sample space Ω . The cumulative sum of the white noise of Figure 5 is a discrete sample sequence of Brownian motion with unit diffusion, $Q(t) = 1$. Three realizations are depicted in Figure 6. Similarly, β indicates Brownian motion, t_n is for time, and ω_i indicates the realization from the sample space. If the diffusion strength $Q(t)$ is increased, the variance in the realizations of the Brownian motions also increases. In other words, the variance or mean squared value of the process grows at a greater rate. The vertical scales in the figures can be adjusted to depict other than Brownian motion with unit diffusion strength.

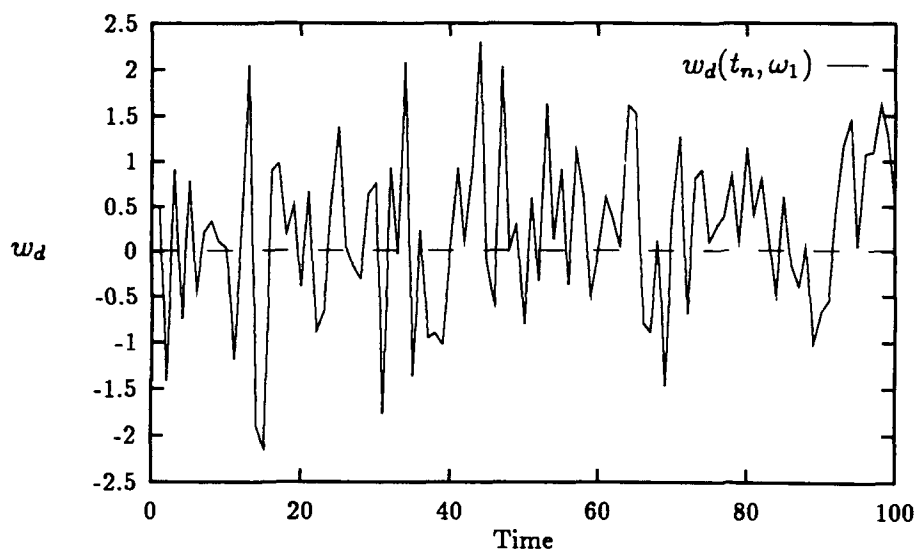


Figure 5. Discrete Sample of White Noise

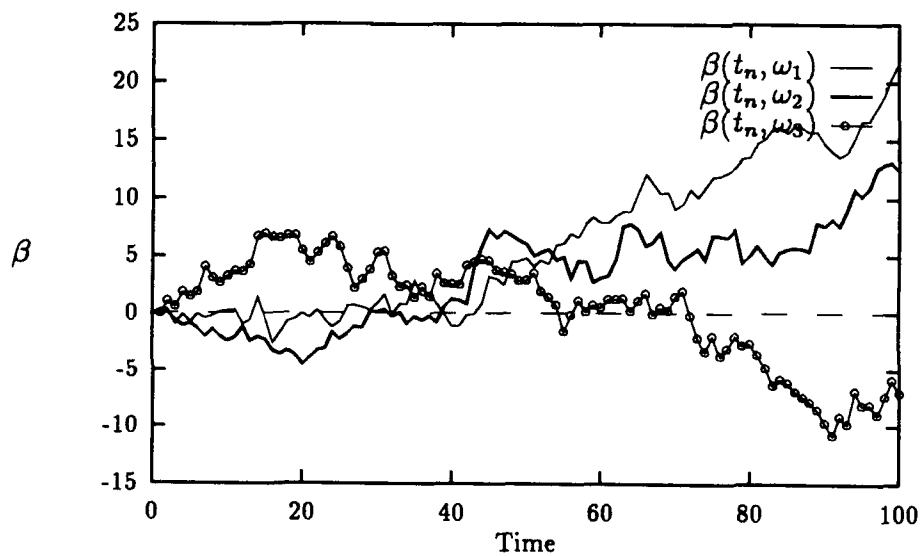


Figure 6. Discrete Samples of Brownian Motion

3.2.2 Dynamics Equation. The general time-varying solution to the linear stochastic differential equation (17) can be shown to be [57:40,164]

$$\mathbf{x}(t) = \Phi(t, t_0)\mathbf{x}(t_0) + \int_{t_0}^t \Phi(t, \tau)\mathbf{G}(\tau)d\beta(\tau)$$

where $\Phi(t, t_0)$ is the state transition matrix from time t_0 to t that satisfies the differential equation and initial conditions:

$$d[\Phi(t, t_0)]/dt = \mathbf{F}(t)\Phi(t, t_0)$$

$$\Phi(t_0, t_0) = \mathbf{I}$$

When the dynamics system matrix $\mathbf{F}(t)$ is a constant matrix, the system is described by a linear time-invariant differential equation, and the transition matrix can be calculated using $\Phi(t, t_0) = \Phi(t - t_0) = e^{\mathbf{F}(t-t_0)}$ or in the Laplace domain $\Phi(s) = [s\mathbf{I} - \mathbf{F}]^{-1}$ [57:41]. Since this research is limited to simulation models with output that attains a stationary steady-state distribution, only time-invariant system models are used.

Since the simulation output is a set of discrete realizations or observations of the stochastic process, for this application an equivalent discrete-time model is useful. Beginning with the continuous-time solution and integrating over time results in the discrete-time version of the dynamics equation.

$$\mathbf{x}(t_{n+1}) = \Phi(t_{n+1}, t_n)\mathbf{x}(t_n) + \mathbf{w}_d(t_n)$$

$$\text{where } \mathbf{w}_d(t_n) = \int_{t_n}^{t_{n+1}} \Phi(t_{n+1}, \tau)\mathbf{G}(\tau)d\beta(\tau)$$

The transition matrix Φ relates the state vector $\mathbf{x}(t_n)$ at one time to the state vector at the next time index $\mathbf{x}(t_{n+1})$. Since the continuous-time dynamics driving noise is assumed to be Brownian motion, the integration or summation of normally-distributed Brownian increments is also normally distributed. Therefore, discrete-time dynamics driving noise

$\mathbf{w}_d(t_n)$ is normally distributed with zero-mean and the following characteristics:

$$\begin{aligned} E[\mathbf{w}_d(t_n)] &= 0 \\ E[\mathbf{w}_d(t_n)\mathbf{w}_d^T(t_n)] &= \int_{t_n}^{t_{n+1}} \Phi(t_{n+1}, \tau) \mathbf{G}(\tau) \mathbf{Q}(\tau) \mathbf{G}^T(\tau) \Phi^T(t_{n+1}, \tau) d\tau = \mathbf{Q}_d(t_n) \\ E[\mathbf{w}_d(t_i)\mathbf{w}_d^T(t_j)] &= 0 \text{ for } t_i \neq t_j \end{aligned} \quad (18)$$

The third characteristic is a result of the Brownian motion having independent increments, which makes the dynamics noise sequence $\{\mathbf{w}_d(t_n)\}$ uncorrelated in time or white.

While discrete-time dynamics equations are often derived from continuous-time relationships, discrete-time dynamics which do not have corresponding continuous-time equations are also useful.

$$\mathbf{x}(t_{n+1}) = \Phi(t_{n+1}, t_n) \mathbf{x}(t_n) + \mathbf{G}_d(t_n) \mathbf{w}_d(t_n) \quad (19)$$

For example, Equation (19) incorporates a $\mathbf{G}_d(t_n)$ that determines which filter-design states the noise affects. This process only has a corresponding continuous process if the resulting dynamics noise covariance matrix $\mathbf{Q}_d(t_n)$ is of full rank. In this research, the formulation in Equation (19) is used. Since only discrete output sequences are considered, the lack of a corresponding continuous relationship has no adverse impact.

Assuming equally-spaced samples and a time-invariant system model, the transition matrix is constant $\Phi(t_{n+1}, t_n) = \Phi$. For application to simulation output, equal spacing can be achieved in terms of simulated time or an index for the output sequence. For example, customers processing through a queuing system can be numbered sequentially for an index [20]. With the time index formulation, the transition matrix represents the relationship between subsequent entities. In addition, since steady-state simulation output is modeled, the dynamics noise $\mathbf{w}_d(t_n)$ is assumed to have constant variance, $\mathbf{Q}_d(t_n) = \mathbf{Q}_d$, and the distribution matrix $\mathbf{G}(t)$ is also assumed to be a constant \mathbf{G} throughout the simulation process.

3.2.3 Measurement Model. In order to apply the Kalman filter, measurements of the system states are necessary. In these applications, the simulation output is used to determine discrete measurements \mathbf{z}_n of the filter-design states $\mathbf{x}(t_n)$. The measurement model indicates the relationship between the system states and the measurements:

$$\mathbf{z}(t_n) = \mathbf{H}(t_n)\mathbf{x}(t_n) + \mathbf{v}(t_n) \quad (20)$$

where \mathbf{z} is the measurement or observation

\mathbf{H} is the measurement or observation matrix

\mathbf{x} is the vector of filter-design system states

\mathbf{v} is the measurement noise

Assume that the measurement noise sequence $\{\mathbf{v}(t_n)\}$ is white (uncorrelated in time), normal, distributed, zero-mean stochastic sequence, such that,

$$\begin{aligned} E[\mathbf{v}(t_n)] &= \mathbf{0} \\ E[\mathbf{v}(t_n)\mathbf{v}^T(t_n)] &= \mathbf{R}(t_n) \\ E[\mathbf{v}(t_i)\mathbf{v}^T(t_j)] &= \mathbf{0} \text{ for } t_i \neq t_j \end{aligned} \quad (21)$$

If the measurement matrix $\mathbf{H}(t_n)$ is not a function of the states $\mathbf{x}(t)$, then the measurement equation is linear. In addition, for steady-state simulation output sequences, the measurement matrix $\mathbf{H}(t_n)$ and the variance of the measurement noise process $\mathbf{R}(t_n)$ are assumed not to change with time. Therefore, these time-dependent matrices are replaced with constant matrices, $\mathbf{H}(t_n) = \mathbf{H}$ and $\mathbf{R}(t_n) = \mathbf{R}$. In addition, the noise sequences $\{\mathbf{w}_d(t_n)\}$ and $\{\mathbf{v}(t_n)\}$ generally are modeled to be uncorrelated with each other.

This section reviews stochastic processes. For a linear stochastic differential equation, the dynamics model and measurement model are shown. Based on these equations, the next section describes the Kalman filter.

3.3 Linear Gaussian Time-Invariant Kalman Filter Algorithm

The discrete-time Kalman filter algorithm is shown for a time-invariant, linear system, with no control inputs, and normally-distributed (Gaussian) zero-mean discrete dynamics noise and measurement noise [57:275]. "Discrete-time" implies that the propagation and measurements updates occur only at set intervals, in contrast to many continuous-time engineering applications. "Linear" implies that the values of the filter-design system states $\mathbf{x}(t_n)$ do not affect the values of the transition matrix Φ , the noise input matrix \mathbf{G} , or the measurement matrix \mathbf{H} . "Time-invariant" means that the matrices Φ , \mathbf{G} , and \mathbf{H} do not change throughout the stochastic process. The covariance matrices of the noises \mathbf{Q}_d and \mathbf{R} are constant since the process variance is assumed stationary.

In this section, the two stages of the Kalman filter, the propagation stage and the measurement update stage, are shown. An example of a Kalman filter is depicted. The section also includes discussions on constant gain Kalman filters and optimality of the filter's state estimates.

3.3.1 Propagation Stage. Two equations comprise the propagation stage. The first relationship determines the propagation through time of the estimated state vector. The second equation propagates the covariance matrix of the state variables through time. The propagation equation takes the best state estimate at the previous time $\hat{\mathbf{x}}(t_{n-1}^+)$ (or the initial estimate $\hat{\mathbf{x}}(t_0)$) and moves it through time by multiplying by the transition matrix Φ . Therefore, the propagation equation is

$$\hat{\mathbf{x}}(t_n^-) = \Phi \hat{\mathbf{x}}(t_{n-1}^+) \quad (22)$$

where $\hat{\mathbf{x}}$ is the estimated state vector

Φ is the transition matrix

This estimate results from computing the conditional expectation of the dynamics model, Equation (19), since the dynamics driving noise $\mathbf{w}_d(t_n)$ has a mean of zero.

The associated covariance matrix $\mathbf{P}(t_n^-)$ of the state estimate $\hat{\mathbf{x}}(t_n^-)$ is calculated with

$$\mathbf{P}(t_n^-) = \Phi \mathbf{P}(t_{n-1}^+) \Phi^T + \mathbf{G}_d \mathbf{Q}_d \mathbf{G}_d^T \quad (23)$$

where \mathbf{P} is the covariance matrix of state estimates $\hat{\mathbf{x}}$

\mathbf{G}_d is the dynamics noise $\mathbf{w}_d(t_n)$ input matrix

\mathbf{Q}_d is the covariance matrix of the discrete dynamics driving noise

These two equations, one for the state estimate and the other for the associated covariance, complete the propagation stage. Each propagation is followed by a measurement update, and then the two-stage cycle begins again.

3.3.2 Measurement Update Stage. After the propagation stage, the second step is the measurement update. The measurement update is actually a static estimation problem of combining two separate sets of information. A state estimate results from the dynamics equation in the propagation stage. The correction to that state estimate is based upon the actual measurement and the measurement model. Both estimates are normally distributed with known covariance matrices. The Kalman filter gain $\mathbf{K}(t_n)$ provides the weighting between the two sets of information about the state.

$$\mathbf{K}(t_n) = \mathbf{P}(t_n^-) \mathbf{H}^T [\mathbf{H} \mathbf{P}(t_n^-) \mathbf{H}^T + \mathbf{R}]^{-1} \quad (24)$$

where \mathbf{H} is the measurement or observation matrix

\mathbf{R} is the measurement covariance noise matrix

\mathbf{K} is the Kalman gain

The measurement update equations, shown below, determine the new state estimates and covariance matrix after the measurement at time t_n is incorporated. The new updated state estimate is

$$\hat{\mathbf{x}}(t_n^+) = \hat{\mathbf{x}}(t_n^-) + \mathbf{K}(t_n) [\mathbf{z}_n - \mathbf{H} \hat{\mathbf{x}}(t_n^-)] \quad (25)$$

where \mathbf{z}_n is the actual measurement at time t_n

The updated state estimate $\hat{\mathbf{x}}(t_n^+)$ is the previous state estimate $\hat{\mathbf{x}}(t_n^-)$ corrected with the Kalman filter gain times the new information provided by the measurement. The new information is obtained from the measurement by taking the actual measurement \mathbf{z}_n and subtracting the best prediction of the measurement before the measurement is received. $\mathbf{H}\hat{\mathbf{x}}(t_n^-)$ is the prediction of the measurement based on the assumed measurement model, Equation (20), and the fact that the measurement noise $\mathbf{v}(t_n)$ has a mean of zero. The differences between the actual measurements and the predicted measurements are the residuals.

The residual,

$$\mathbf{r}(t_n) = \mathbf{z}_n - \mathbf{H}\hat{\mathbf{x}}(t_n^-) \quad (26)$$

can be viewed as the new information contained in the measurement. Under the assumptions of linear, Gaussian (normally-distributed noise terms), time-invariant models with known system matrices (Φ , \mathbf{G}_d , \mathbf{Q}_d , \mathbf{H} , and \mathbf{R}), the residuals are normally distributed with a mean vector of $E[\mathbf{r}(t_n)] = \mathbf{0}$ and covariance matrix of $E[\mathbf{r}(t_n)\mathbf{r}^T(t_n)] = \mathbf{H}\mathbf{P}(t_n^-)\mathbf{H}^T + \mathbf{R}$. In addition, the residual sequence $\{\mathbf{r}(t_n)\}$ is a white (uncorrelated in time) sequence [57:228-229]. The new information contained in the residual $\mathbf{r}(t_n)$ may be substituted into Equation (25) to get a simpler form of the state updated equation.

$$\hat{\mathbf{x}}(t_n^+) = \hat{\mathbf{x}}(t_n^-) + \mathbf{K}(t_n)\mathbf{r}(t_n) \quad (27)$$

The Kalman filter residuals, from Equation (26), are critical in the simulation applications of the Kalman filter. Besides representing the new information in each successive measurement, the residuals can be thought of as the error in the dynamics equation prediction as projected onto the measurement space. Thus, one approach to estimating unknown parameters in the dynamics and measurement equations is selecting parameters which minimize the magnitude of the residuals. The common technique is to use least squares estimation, which minimizes the sum of the squared residuals. If the steady-state model of the simulation output can be approximated with an appropriate linear model, the residuals should constitute a white, normally distributed, zero mean sequence. The Kalman filter residuals may provide

an indication of the adequacy of the estimated model's fit to the data being processed.

Along with an updated state estimate, an updated covariance matrix is also necessary. Since the Kalman filter gain $\mathbf{K}(t_n)$ is the weight based on comparing the variances of the estimate from the propagation stage and the measurement model, it stands to reason that this same gain is used to determine the reduction in variance resulting from incorporating the measurement information:

$$\mathbf{P}(t_n^+) = \mathbf{P}(t_n^-) - \mathbf{K}(t_n)\mathbf{H}\mathbf{P}(t_n^-) \quad (28)$$

In the scalar case, since at worst the measurement contributes no new information, the updated variance is always less than or equal to the propagated variance, $P(t_n^+) \leq P(t_n^-)$. Then during the propagation phase, the variance increases to $P(t_{n+1}^-)$ because of the uncertainty induced by the dynamics driving noise.

The Kalman filter gain $\mathbf{K}(t_n)$ weights the information provided by the dynamics equation and the measurement information according to their variances. As the terms of \mathbf{Q}_d , and hence $\mathbf{K}(t_n)$, increase, the measurement information is weighted heavier and the dynamics model prediction is weighted less. Conversely, as the terms of \mathbf{R} increase, and thus elements of $\mathbf{K}(t_n)$ decrease, the measurement information is weighted less.

For a moment, consider a scalar state vector $x(t_n)$. If measurements of this state are available, the measurement matrix $H = 1$. Under these conditions, the Kalman filter gain simplifies to $K(t_n) = \frac{P(t_n^-)}{P(t_n^-) + R}$. If the measurements are perfect, the measurement noise variance is zero, $R = 0$. With $R = 0$, the Kalman filter gain is one, $K = 1$, and thus $\hat{x}(t_n) = z_n$. Therefore, the update equation completely disregards the prediction based on the dynamics equation of the propagation stage, and the updated estimate is based entirely on the measurement model prediction. This is logical since the measurements of the state are perfect.

At the opposite end of the spectrum, if the measurement is so poor that it provides no additional information, the measurement noise variance equals infinity, $R = \infty$. Under

these conditions, the Kalman filter gain is zero, $K = 0$. Thus, the update step totally disregards the worthless measurement, and the updated prediction would be based entirely on the propagation equations, $\hat{x}(t_n^+) = \hat{x}(t_n^-)$.

In actual practice, both the propagation state estimate and the measurement model state estimate are combined for the best estimate. The weights applied are determined by the Kalman filter gain which accounts for the different quality of their information by means of their variances.

Another important observation is that the covariances $P(t_n^-)$ and $P(t_n^+)$ do not depend on the actual measurements. Inspection of Equations (23), (24) and (28) shows the covariance matrices $P(t_n^-)$ and $P(t_n^+)$ are unaffected by the actual measurements z_n . In fact, with constant matrices, Φ, G_d, Q_d, H and R , the covariance matrices, $P(t_n^-)$ and $P(t_n^+)$, and hence the Kalman filter gain vector $K(t_n)$, attain constant matrices, P^-, P^+ , and K , as the influence of initial values of $P(t_0)$ decays. Therefore, these values can be precomputed and stored prior to the actual running of the Kalman filter.

Besides the variance estimates being independent of the actual measurements, the magnitude of the constant gain K depends only on the relationship between Q_d and R , not their actual values. In the case of scalar noises, the ratio $\frac{Q_d}{R}$ determines the gain K . This indeterminacy complicates simultaneous estimation of both Q_d and R .

3.3.3 Kalman Filter One Dimensional Example. In order to illustrate the use of the Kalman filter, a simple scalar example is shown. The initial estimate, variance, and the system parameters are as follows:

Initial state estimate $\hat{x}_0 = 100$	Initial variance $P_0 = 1$
Transition matrix $\Phi = 0.9$	Dynamics noise variance $Q_d = 5.2$
Measurement matrix $H = 1$	Measurement noise variance $R = 3.0$
Dynamic noise input matrix $G_d = 1$	

The initial state estimate is normally distributed with mean of \hat{x}_0 and variance of P_0 .

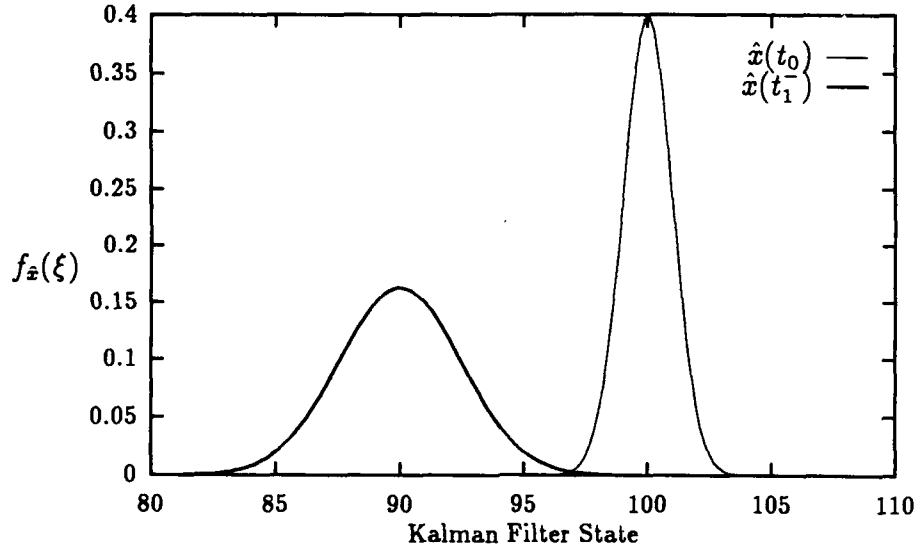


Figure 7. Propagation Through Time

The propagation stage moves the normal probability density of the initial state estimate forward in time to t_1 :

$$\hat{x}(t_1^-) = \Phi \hat{x}(t_0) = 0.9(100) = 90$$

$$P(t_1^-) = \Phi P(t_0) \Phi^T + G_d Q_d G_d^T = (0.9)^2(1.0) + (1)5.2(1) \approx 6.0$$

The propagation stage maintains the normal probability density with a mean of $\hat{x}(t_1^-)$ and a variance of $P(t_1^-)$. Figure 7 depicts the move of the normal distributed state estimates from a mean of 100 at time t_0 to a mean of 90 at time t_1 with an associated increase in variance due to the dynamics driving noise.

The measurement update at t_1 incorporates the information contained in a measurement received at that time. For this example, let the measurement be $z_1 = 84$:

$$K(t_1) = \frac{P(t_1^-)H}{HP(t_1^-)H+R} = \frac{6.0(1)}{1(6.0)1+3} = \frac{2}{3}$$

$$\hat{x}(t_1^+) = \hat{x}(t_1^-) + K(t_1)[z_1 - H\hat{x}(t_1^-)] = 90 + \frac{2}{3}[84 - 1(90)] = 86$$

$$P(t_1^+) = P(t_1^-) - K(t_1)HP(t_1^-) = 6.0 - \frac{2}{3}(1)6.0 = 2.0$$

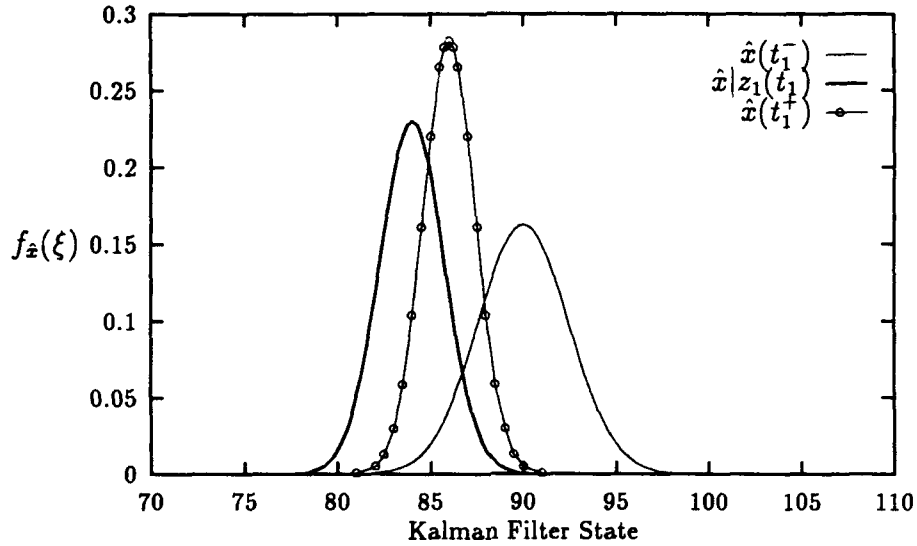


Figure 8. Measurement Update

From the measurement model, $z(t_n) = Hx(t_n) + v(t_n)$ with $H = 1$ and an actual measurement z_n , an estimate of the state conditioned only on the new measurement $\hat{x}|z_n(t_n)$ equals $z_n - v(t_n)$. Although the measurement noise $v(t_n)$ is unknown, it is normally distributed with a zero mean and a variance of $R(t_n)$. Therefore, the state estimate conditioned on the new measurement at time t_n , $\hat{x}|z_n(t_n)$, is also normally distributed with a mean value corresponding to the measurement z_n and the variance of the measurement noise $R(t_n)$.

The Kalman filter gain $K(t_n)$ provides the appropriate weight to combine these two normal probability distributions. The propagated distribution with a mean of $\hat{x}(t_n^-)$ receives a weight of $1 - K(t_n)$, and the distribution conditioned on the new measurement with mean of $\hat{x}|z_n(t_n)$ is weighted by $K(t_n)$. Since this is a linear combination of two normally-distributed random variables, the resulting estimate $\hat{x}(t_n^+)$ is also normally distributed. Furthermore, since $K(t_n)$ is bounded between zero and one, the resulting distribution has a mean between the previous two means. Figure 8 shows the distributions of the propagated state estimate $\hat{x}(t_1^-)$, the state estimate conditioned on the new measurement $\hat{x}|z_1(t_1)$, and the updated state estimate $\hat{x}(t_1^+)$. The updated state estimate has smaller variance than either the propagated

state estimate or the state estimate conditioned on the measurement.

The recursive nature of the Kalman filter algorithm is that it repeats the steps again beginning with the updated state estimates. The two steps of propagation and update are repeated for each observation.

3.3.4 Constant Kalman Filter Gain. As mentioned previously, the Kalman filter gain $\mathbf{K}(t_n)$ and the covariance matrices $\mathbf{P}(t_n^-)$ and $\mathbf{P}(t_n^+)$ are not dependent upon the actual measurements. Actually, these matrices are functions of the initial estimate $\mathbf{P}(t_0)$, the covariance of the dynamics driving noise $\mathbf{Q}_d(t_n)$, and the measurement noise covariance $\mathbf{R}(t_n)$. Since \mathbf{Q}_d and \mathbf{R} are constant (as well as Φ , \mathbf{G}_d , and \mathbf{H}) in these applications, $\mathbf{K}(t_n)$, $\mathbf{P}(t_n^-)$, and $\mathbf{P}(t_n^+)$ attain steady-state values as the contribution of $\mathbf{P}(t_0)$ decays. These steady-state values can be determined from the Kalman filter equations by setting the variance matrices to constant matrices. The constant variance version of Equation (23) is

$$\mathbf{P}^- = \Phi \mathbf{P}^+ \Phi^T + \mathbf{G}_d \mathbf{Q}_d \mathbf{G}_d^T \quad (29)$$

From Equation (24), the steady-state Kalman filter gain is

$$\mathbf{K} = \mathbf{P}^- \mathbf{H}^T [\mathbf{H} \mathbf{P}^- \mathbf{H}^T + \mathbf{R}]^{-1} \quad (30)$$

and from Equation (28),

$$\mathbf{P}^+ = \mathbf{P}^- - \mathbf{K} \mathbf{H} \mathbf{P}^- \quad (31)$$

Substituting for \mathbf{P}^+ and \mathbf{K} into Equation (29) results in

$$\mathbf{P}^- - \Phi \mathbf{P}^- \Phi^T + \Phi \mathbf{P}^- \mathbf{H}^T [\mathbf{H} \mathbf{P}^- \mathbf{H}^T + \mathbf{R}]^{-1} \mathbf{H} \mathbf{P}^- \Phi^T - \mathbf{G}_d \mathbf{Q}_d \mathbf{G}_d^T = 0 \quad (32)$$

This equation is called an algebraic Riccati equation because of its form. For the previous one-dimensional example, the progression of these values is in Table 5 for calculations to two decimal places. The state estimate variances and the Kalman filter gain rapidly converge to

Table 5. Steady-State Variances and Gain

t_n	$P(t_n^-)$	$K(t_n)$	$P(t_n^+)$
0			1.00
1	6.01	0.67	1.98
2	6.81	0.69	2.11
3	6.91	0.70	2.07
4	6.88	0.70	2.06
5	6.87	0.70	2.06
\vdots	\vdots	\vdots	\vdots

their steady-state values.

Using the steady-state matrices, P^- , P^+ , K , the two-step Kalman filter algorithm simplifies to only two equations, Equations (22) and (25) with a constant Kalman gain K . The covariance matrices are not calculated since their steady-state values are used for all time. For the simple example shown previously, the steady-state values are used to generate the Kalman filter state estimates in Figure 9.

After applying the Kalman Filter to the simulation output data, the estimates of the filter-design state vector, the residual vectors, and also the covariance matrices associated with these vectors are available. In addition, the covariance over time or the autocovariance matrix $P_{xx}(t_1, t_2)$ is also available [57:166]:

$$P_{xx}(t_1, t_2) = \Phi P_{xx}(t_1, t_1) = \Phi P$$

This additional information may be used to improve simulation output analysis techniques.

3.3.5 Kalman Filter Optimality. Under the following assumptions, Kalman filter state estimates are guaranteed to be optimal by essentially any logical criteria, such as any symmetric cost function [57:231-236].

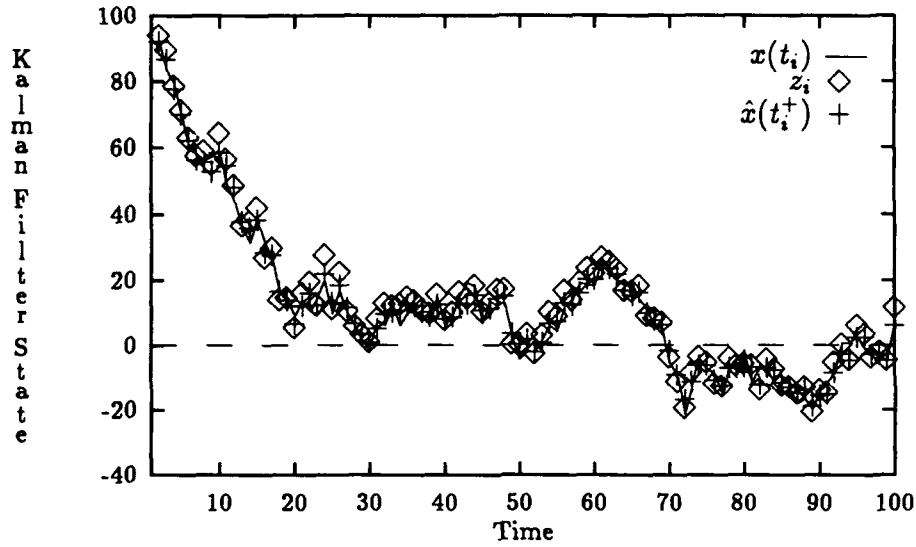


Figure 9. Steady-State Kalman Filter

- The dynamics equation is a linear stochastic differential equation, and the matrices $F(t)$ and $G(t)$, or equivalently $\Phi(t_{n+1}, t_n)$ and $G_d(t)$, are known. The measurement model is linear, and the matrix $H(t_n)$ is known.
- The dynamics driving noise is Brownian motion with known strength $Q(t_n)$, or equivalently, the discrete-time white dynamics noise $w_d(t_n)$ has a known covariance matrix $Q_d(t_n)$. The measurement noise is uncorrelated normally-distributed random sequence with known covariance matrix $R(t_n)$.
- The initial state estimates are either known or normally distributed with a known mean \hat{x}_0 and covariance matrix P_0 .

Since Brownian motion has normally-distributed independent increments, all of the random variable inputs are known or normally distributed. In addition, since the stochastic differential equation is linear, all of the state estimates are linear combinations of jointly normally-distributed random variables. Therefore, all of the state estimates are also normally-distributed random variables.

The advantage of working with normally-distributed random variables is that the odd central moments beyond the first one, the mean, are all zero. For example, the third moment which measures skewness is zero since the normal distribution is symmetric. In addition, the even central moments are all functions of the second central moment, the variance. Therefore, the first two central moments, the mean and the variance, completely specify the normal probability distribution.

Furthermore, the mean, mode, and median are all the same point for the normal distribution. Since most of the optimality criteria select one of these points as the optimal estimate, under the above assumptions, the Kalman filter estimates are optimal by essentially all meaningful criteria. According to Maybeck [57:232], the Kalman filter state estimate $\hat{\mathbf{x}}(t_i^+)$ is the optimal Bayesian estimate because it is the mean of the probability distribution of the state conditioned on the measurements, $f_{\mathbf{x}(t_i)|\mathbf{Z}(t_i)}(\xi|\mathbf{Z}_i)$, where \mathbf{Z}_i is the entire measurement history available at time t_i . “By virtue of being the conditional mean, $\hat{\mathbf{x}}(t_i^+)$ is also the minimum mean square error (MMSE) estimate [57:232].” In addition, by being the conditional mean of a symmetric distribution, $\hat{\mathbf{x}}(t_i^+)$ minimizes any symmetric cost function criteria [57:232]. While the Bayesian estimate is the conditional mean of $f_{\mathbf{x}(t_i)|\mathbf{Z}(t_i)}(\xi|\mathbf{Z}_i)$, the mode of this distribution is called the maximum a posteriori (MAP) estimate. Similarly, the classical maximum likelihood estimate (MLE) is the mode of $f_{\mathbf{Z}(t_i)|\mathbf{x}(t_i)}(\mathbf{Z}_i|\xi)$ [57:235]. Besides being the MAP estimate, “ $\hat{\mathbf{x}}(t_i^+)$ is the maximum likelihood estimate (MLE) of $\mathbf{x}(t_i)$ if $\mathbf{P}_0 = \infty\mathbf{I}$, i.e. if $\mathbf{P}_0^{-1} = \mathbf{0}$, and converges asymptotically to the MLE if $\mathbf{P}_0^{-1} \neq \mathbf{0}$.” Maybeck [57:235] further states that $\hat{\mathbf{x}}(t_i^+)$ is the minimum variance unbiased linear estimate even if the normal distribution assumption is removed from all noise inputs.

3.4 System Identification

Before applying a Kalman filter to simulation output, the system dynamics and measurement equations must be determined. For engineering applications, these equations are developed by aggregating the effect of subsystem components and empirical testing. Since discrete-event simulations are used in applications where no analytically tractable solution

exists, the dynamics and measurement equations must be deduced from the output sequence.

Assume the discrete stochastic process of simulation output, which has attained a stationary steady-state distribution, can be represented by a linear time-invariant discrete system with normally-distributed noise inputs. The time-invariant version of the dynamics model, Equation (19),

$$\mathbf{x}(t_n) = \Phi \mathbf{x}(t_{n-1}) + \mathbf{G}_d \mathbf{w}_d(t_n) \quad (33)$$

is determined by the constant transition matrix Φ , the constant dynamic noise input matrix \mathbf{G}_d , and the time-invariant covariance matrix \mathbf{Q}_d of the discrete white normally-distributed zero-mean dynamics driving noise $\mathbf{w}_d(t_n)$. Similarly, the time-invariant version of the measurement or observation model, Equation (20),

$$\mathbf{z}(t_n) = \mathbf{H} \mathbf{x}(t_n) + \mathbf{v}(t_n) \quad (34)$$

is determined by specifying the constant measurement matrix \mathbf{H} and the constant covariance matrix \mathbf{R} of the discrete white normally-distributed zero-mean measurement noise $\mathbf{v}(t_n)$. Therefore to specify the system equations completely, Φ , \mathbf{G} , \mathbf{Q}_d , \mathbf{H} and \mathbf{R} must be estimated. However, due to this estimation, this Kalman filter application is no longer linearly independent of the observations and optimality cannot be guaranteed.

Four potential approaches are considered to determine the appropriate system equation and estimate the necessary parameters. First, a range of structural forms and parameterizations can be evaluated by the maximum likelihood estimation (MLE) procedure developed by Akaike [6, 4, 5, 7, 8]. Second, multiple model adaptive estimation (MMAE) [48, 49, 58, 60] can be applied with various classes of models. Third, transfer functions can be developed to relate past observations to future observations. Finally, the structure of the model can be imposed by *a priori* considerations, and the necessary parameters can be estimated by least squares, MLE or MMAE. Each of these four approaches to determine the structure will be presented in the following sections. However, only the *a priori* formulation is selected, applied, and tested.

3.4.1 Akaike's Estimation Methodology. Several researchers demonstrate the equivalence of some common types of model formulations. Akaike [4] proves that there is no difference between a Markovian representation of an output sequence from a stationary stochastic process and an autoregressive and moving average (ARMA) representation. Cooper and Wood [16] show equivalence between ARMA and state-space representation without control inputs. Hence, if an ARMA model can be fit to simulation output, an equivalent state-space model can be developed. The reverse may not be true since state-space formulations allow for measurement noise which is not included in ARMA models.

While engineering models are built using known relationships, time series models are estimated typically from the realized sequence. Box and Jenkins [11] present a methodology to fit ARMA models to univariate data. Two limitations in their methodology are that subjective assessments are required and that multivariate extensions are difficult [8]. Overcoming the multivariate limitation, Charnes [12] extends the use of time series models for simulation output analysis. He demonstrates that vector autoregressive moving average (VARMA) models of specific forms can be used to improve the estimation of combined confidence regions for simulation system parameters. Charnes' results demonstrate that simulation output sequences can be successfully modeled with time series formulations, and hence, also with state-space models.

Akaike [4] proposes the following maximum likelihood procedure for stationary normally-distributed processes with a Markovian representation. Markovian representation ensures a finite vector representation. First, the multivariate normal probability density function is

$$f(\xi) = (2\pi)^{-S/2} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} [\xi - \mu]^T \Sigma^{-1} [\xi - \mu] \right\}$$

where ξ is a realization of dimension S , μ is the mean vector, and Σ is the covariance matrix. Since the natural logarithm is a strictly monotonic increasing function, the maximum of the logarithm of a function is the same as the maximum of the original function. Letting

$L = f(\xi)$, the natural logarithm of the likelihood equals

$$\ln(L) = -\frac{S}{2} \ln(2\pi) - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} [\xi - \mu]^T \Sigma^{-1} [\xi - \mu]$$

Define actual residuals \mathbf{r}_n as the measurements \mathbf{z}_n minus the predicted measurements $\mathbf{H}\hat{\mathbf{x}}(t_n^-)$, as shown in Equation (26). Using the whiteness of the residuals, their joint density function can be written as a product of the normally distributed marginal densities of the individual residuals. Incorporating the actual residuals and their associated covariance matrix, the natural logarithm of the likelihood function for N observations $\{\mathbf{z}_n\}$ is:

$$\ln(L) = -\frac{SN}{2} \ln(2\pi) - \frac{1}{2} \sum_{n=1}^N \ln |\mathbf{H}\mathbf{P}^- \mathbf{H}^T + \mathbf{R}| - \frac{1}{2} \sum_{n=1}^N \mathbf{r}_n^T [\mathbf{H}\mathbf{P}^- \mathbf{H}^T + \mathbf{R}]^{-1} \mathbf{r}_n \quad (35)$$

where S is the dimension of the measurement vector \mathbf{z}_n

N is the number of observations

\mathbf{H} is the measurement matrix

\mathbf{P}^- is the covariance matrix of the state estimates prior to the measurement update

\mathbf{R} is the covariance matrix of the measurement noise

\mathbf{r}_n is the residual vector for the n th observation

For each set of parameters considered, the data must be processed through the Kalman filter because the likelihood function assumes a complete sequence of Kalman filter state estimates. This likelihood function can be maximized by employing a nonlinear optimization routine where each evaluation requires processing all of the observations through a Kalman filter.

The formulation can be varied by including more or less of the past observations in the observation vector. For a specified dimension, the matrices, $\Phi, \mathbf{G}_d, \mathbf{H}, \mathbf{Q}_d, \mathbf{R}$, which maximized this function are the Maximum Likelihood Estimation (MLE) parameters. MLE estimates the parameters without any explicit penalty for increasing the number of parameters by using a larger system-state matrix. In contrast, Akaike's [4, 8] technique adds a penalty for using a system-state vector with larger dimensions and more parameters to

estimate. Akaike defines his information criterion, the AIC, as

$$\begin{aligned} \text{AIC} = & -2 \ln (\text{maximum value of the likelihood function}) \\ & + 2 (\text{number of independently adjusted parameters within the model}) \end{aligned}$$

The first term is a penalty for "badness of fit" of the model, and the second term penalizes increased unreliability for additional parameter estimation [4]. When models of the same dimension, and hence the same number of parameters, are compared, the addition of a constant does not change the minimum of the AIC. Therefore, for models of the same dimension, the technique results in the classical MLE methodology. For models of different dimensions, the best choice of model is the one that minimizes the AIC. Akaike refers to the selected model as the minimum information criteria estimate, MAICE. Akaike summarizes the value of his procedure as follows:

By the introduction of MAICE the problem of statistical identification is formulated explicitly as a problem of estimation and the need of the subjective judgment required in the hypothesis testing procedure for the decision on the levels of significance is eliminated completely. [5]

Because of the extensive numerical computations necessary to search all potential models of various dimensions and parameterizations using the AIC, Akaike [4] says the feasibility of this procedure is almost entirely dependent on a good initial guess at the dimension of the system and selecting a corresponding basis of the state space. He further states that determination of the Markovian or ARMA representation proceeds in two steps, first selection of the structure and then determination of the parameters.

Akaike [6] further recommends that canonical correlation analysis for the process can provide an initial guess of the structure and the parameters to be used for the initialization of the maximum likelihood procedure. Akaike [7] says that, for two sets of random variables, one composed of the past and present values and the other composed of the present and future values, the two sets of canonical variables with positive correlation form the minimal

information interface between the past and the future of the process. From this minimal information interface, the dimension of the transition matrix can be determined.

Green [28:261] describes the objective of canonical correlation as finding the linear composite variables for each of two sets such that the highest possible correlation is attained between the linear composites. Subsequent linear composites are selected such that they are uncorrelated with previous linear composites and the highest conditional correlation is attained. Each successive pair of composites between the two sets exhibits decreasing correlation.

Let \mathbf{x} represent one variable set with an associated vector of weights \mathbf{a} . A new set of variables \mathbf{x}^* is determined by $\mathbf{a}^T \mathbf{x}$. Similarly, the other set of variables \mathbf{y} and another vector of weights \mathbf{b} defines a new set of variables \mathbf{y}^* . The objective of canonical correlation is to specify \mathbf{a} and \mathbf{b} such that correlation between \mathbf{x}^* and \mathbf{y}^* is maximized. A standard multivariate result is that the \mathbf{a} and \mathbf{b} which maximizes the correlation between the two sets are the solutions to the eigenvalue problems:

$$(\Sigma_{xx}^{-1} \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} - \lambda \mathbf{I}) \mathbf{a} = 0 \text{ and } (\Sigma_{yy}^{-1} \Sigma_{yx} \Sigma_{xx}^{-1} \Sigma_{xy} - \lambda \mathbf{I}) \mathbf{b} = 0$$

where Σ are the respective covariance or cross-variance matrices, and the largest eigenvalue λ is the squared canonical correlation coefficient [18:341-342].

For a time-invariant system, Akaike [6] defines a realization of the Φ matrix of minimal dimension as a *minimal realization*. Akaike [8:68-70] presents the steps for determining the minimal realization using the MAICE. His process begins by developing two sets of observations for canonical analysis. As he processes through the data, the p past vectors plus the present vector is an observation in the first set of variables. In addition, at each data point, the present vector plus the p future vectors is the corresponding observation in the second set. The process is repeated with increasing p . The number of vectors in the spanning set of canonical variates is the dimension p of the system. This p is equivalent to the number of nonzero canonical variates [8:60], and the resulting model is equivalent to an

ARMA($p, p-1$) model [27:7-8]. Akaike's system identification process iterates between minimization of the AIC given the model dimension and canonical correlation with an additional lagged vector [8:68-70].

Cooper and Wood [17] modify Akaike's procedure by including the present vector in only the future set rather than both the past and the future sets. Goodrich and Stellwagen state in the Forecastmaster software documentation [27:7-9] that the advantage of this modification is that all of the canonical variates can be identified in one step rather than in a stepwise procedure. An ARMA(p, p) model results, and the parameters can be estimated by maximum likelihood techniques using a nonlinear optimization routine such as Gauss-Newton. The parameter estimates which are small compared to their standard error are set to zero. Porter [76] uses Forecastmaster [27] in a feasibility study of applying the Kalman filter to simulation output. He shows that, for waiting times from an $M/M/1$ queue, a two state Kalman filter is specified. Tsay and Tiao [99] and Tsay [98] improve Akaike's methodology by eliminating the requirement for the canonical analysis step to specify models only of order ARMA(p, p).

Schwarz [89] recommends a similar procedure except that Akaike's information criteria (AIC) is replaced with a Bayesian information criteria (BIC):

$$\begin{aligned} \text{BIC} = & -2 \ln(\text{maximum value of the likelihood function}) \\ & + (\text{number of independently adjusted parameters}) \end{aligned}$$

This technique selects the model that is a posteriori most probable. Quantitatively both procedures apply the principle of parsimony [11:17-18] in model building. The Schwarz technique leans toward a lower-dimension model [89]. Neftci [72] makes a comparison between the models resulting from using the AIC and the BIC for economic time series. Applying both criteria to eight economic time series, Neftci fit univariate ARMA models. In every case, Schwarz's BIC specifies a lower-dimension model. Neftci reports that the AIC selected models with over nine parameters on average, whereas the BIC choose models which average less than five parameters.

In concluding this section, Akaike's procedure or one of the derivatives is a possibility for identifying a dynamics model and a measurement equation for simulation output. However, due to the computational requirements to evaluate each set of parameters to determine the MLE, this approach does not appear well suited for this application.

3.4.2 Multiple Model Adaptive Estimation (MMAE). Another potential method for system identification is Multiple Model Adaptive Estimation [34, 48, 49, 58, 60, 61, 62, 63]. MMAE simultaneously estimates uncertain filter-design structural parameters and state estimates. MMAE basically approximates the unknown parameter space with discrete points and runs a Kalman filter at each discretized point in parameter space. The MMAE estimation is accomplished by monitoring the residuals for each of the Kalman filters in the bank.

In contrast to MMAE, the unknown state parameters can be augmented to the state vector and estimated along with the states using a nonlinear estimation technique [58:Chapter 12]. While a nonlinear approach requires approximations, Lainiotis [49] summarizes the relationship between the nonlinear estimation of the state estimate at time t conditioned on the available measurement history, $\hat{\mathbf{x}}(t|t, t_0)$ in his notation, and MMAE as follows:

The optimal *nonlinear* filter constitutes an *exact* decomposition or partitioning of the nonlinear filter for $\hat{\mathbf{x}}(t|t, t_0)$ into a set of much simpler *linear* elemental filters, namely, the Kalman-Bucy filters, which are, moreover, completely *decoupled* from each other. [49]

In addition, MMAE can be used to determine the necessary dimension of the filter-design state vector [48].

Besides determining the system dimension, MMAE can be used to estimate unknown parameters. Let \mathbf{a} denote a vector of the parameters to be estimated. The continuous range of values for \mathbf{a} are discretized into L representative sets of values, \mathbf{a}_i . The discretization of \mathbf{a} can be thought of as a "grid".

The discretization typically is determined by the effect of varying the particular parameter. The *generalized ambiguity function* is defined as the average value of the likelihood

function upon which the estimator of \mathbf{a} is based, and it can provide insight into determining the appropriate number of discrete levels needed for unknown parameters. The generalized ambiguity function is given by Maybeck [58:97-99] where θ is some function of the estimated variables and \mathcal{L}_i represents a realization of the measurement history $\mathbf{Z}(t_i)$,

Let $L[\theta(t_i), \mathcal{L}_i]$ denote that likelihood function, and then the ambiguity function $\mathcal{A}(\cdot, \cdot)$ is defined as the scalar function such that

$$\mathcal{A}(\theta, \theta_t) \equiv \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} L[\theta, \mathcal{L}_i] f_{\mathbf{Z}(t_i)|\theta(t_i)}(\mathcal{L}_i|\theta_t) d\mathcal{L}_i$$

where θ is some value of the estimated variables at time t_i and θ_t is the true, but unknown, value of these quantities.

The curvature of the ambiguity function in the vicinity of the true parameters indicates the preciseness with which the maximum likelihood estimate can be discerned [57:97]. If the ambiguity function has little curvature, meaning it is relatively flat, the parameter effect is not easily discernible, and therefore, a coarse grid for that parameter ought to be sufficient. In contrast, if the ambiguity function has a distinct peak, small variations in the parameter should be discernible in the magnitude of the residuals. Thus, a relatively fine discretation is appropriate. In order to evaluate the ambiguity function, a sensitivity analysis is conducted [57:325-341]. In this analysis, the "truth model" is a detailed model with assumed true parameters. Unlike many engineering applications, most discrete-event simulations do not provide an adequate "truth model".

After the discretization of the parameter space is complete, let the probability that the unknown set of parameters \mathbf{a} assumes the set of L discrete values \mathbf{a}_l conditioned on the measurement history up to time t_n be $p_l(t_n) = \text{Prob}(\mathbf{a} = \mathbf{a}_l | \mathbf{Z}(t_n) = \mathbf{Z}_n)$. Assuming the possible values for \mathbf{a} are limited to the L discrete parameter vectors \mathbf{a}_l , the probability is calculated [61] as

$$p_j(t_n) = \frac{f_{\mathbf{Z}(t_n)|\mathbf{a}, \mathbf{Z}(t_{n-1})}(\mathbf{z}_n | \mathbf{a}_j, \mathbf{Z}_{n-1}) \cdot p_j(t_{n-1})}{\sum_{l=1}^L f_{\mathbf{Z}(t_n)|\mathbf{a}, \mathbf{Z}(t_{n-1})}(\mathbf{z}_n | \mathbf{a}_l, \mathbf{Z}_{n-1}) \cdot p_l(t_{n-1})} \quad (36)$$

The Bayesian minimum mean squared error state estimate, the sum of the L discrete state estimates weighted by their associated probabilities, is the MMAE estimated state vector:

$$\hat{\mathbf{x}}(t_n^+) = \sum_{l=1}^L \hat{\mathbf{x}}_l(t_n^+) \cdot p_l(t_n) \quad (37)$$

Each $\hat{\mathbf{x}}_l(t_n^+)$ is calculated using a discrete-time Kalman filter with the associated \mathbf{a}_l parameters. With time-invariant matrices, the probabilities are calculated using the residuals $\mathbf{r}_l(t_n) = \mathbf{z}_n - \mathbf{H}_l \hat{\mathbf{x}}_l(t_n)$ and their covariance matrices $\mathbf{A}_l = \mathbf{H}_l \mathbf{P}_l^- \mathbf{H}_l^T + \mathbf{R}_l$. Since these residuals are jointly normally distributed,

$$f_{\mathbf{z}(t_n)|\mathbf{a}, \mathbf{Z}(t_{n-1})}(\mathbf{z}_n | \mathbf{a}_l, \mathbf{Z}_{n-1}) = (2\pi)^{-S/2} |\mathbf{A}_l|^{-1/2} \exp \left\{ -\frac{1}{2} \mathbf{r}_l^T(t_n) \mathbf{A}_l^{-1} \mathbf{r}_l(t_n) \right\} \quad (38)$$

where S is the dimension of the measurement vector \mathbf{z}_n . Since the residual $\mathbf{r}_l(t_n)$ for the “best” set of parameters \mathbf{a}_l should be relatively small, the “best” set of parameters is assigned a high probability by the preceding $p_l(t_n)$ computation. Similarly, the residual for a “mismatched” model should be relatively large and the associated probability is small [60].

Maybeck [58:131] derives the covariance matrix for the random vector of the updated MMAE state vector $\hat{\mathbf{x}}(t_n^+)$ as

$$\mathbf{P}(t_n^+) = \sum_{l=1}^L p_l(t_n) \{ \mathbf{P}_l(t_n^+) + [\hat{\mathbf{x}}_l(t_n^+) - \hat{\mathbf{x}}(t_n^+)] [\hat{\mathbf{x}}_l(t_n^+) - \hat{\mathbf{x}}(t_n^+)]^T \}$$

Similarly, from Maybeck [58:132-133], the conditional mean for the unknown vector of system parameters \mathbf{a} at time t_n is

$$\hat{\mathbf{a}}(t_n) = E\{\mathbf{a} - \hat{\mathbf{a}}(t_n) | \mathbf{Z}(t_n) = \mathbf{Z}_n\} = \sum_{l=1}^L \mathbf{a}_l \cdot p_l(t_n) \quad (39)$$

and an indication of the precision of the estimate $\hat{\mathbf{a}}(t_n)$ is given by the conditional covariance matrix of \mathbf{a} :

$$\Sigma_{\mathbf{a}|\mathbf{Z}_n} = E\{[\mathbf{a} - \hat{\mathbf{a}}(t_n)][\mathbf{a} - \hat{\mathbf{a}}(t_n)]^T | \mathbf{Z}(t_n) = \mathbf{Z}_n\} = \sum_{l=1}^L [\mathbf{a}_l - \hat{\mathbf{a}}(t_n)][\mathbf{a}_l - \hat{\mathbf{a}}(t_n)]^T p_l(t_n) \quad (40)$$

In the recursive calculations of $p_l(t_n)$, if any of the $p_l(t_n)$ becomes zero, it remains zero thereafter. To prevent a set of parameters from getting discarded prematurely, the probabilities typically are given a lower bound which depends on the number of discrete points in the parameter space [60]. This lower bound permits the parameters to continue to adapt throughout the observation set.

In contrast to using the Bayesian probabilistic weighted average of each filter, Maybeck and Stevens [63] test two other techniques for determining the state estimate. The first alternative is the Maximum A Posteriori (MAP) version of MMAE, which simply uses the state estimate of the filter with the highest computed probability. This prevents including the weighted average of the filters with probabilities at the lower bound. A middle ground between the Bayesian and MAP approach is to include within the weighted average, only those filters with probabilities over a certain threshold.

The second alternative proposed by Maybeck and Stevens [63] is to modify the probability calculation. They note that the leading term before the exponential term in the probability density function, Equation (38), scales the area under the curve to one. In the case where the residuals \mathbf{r}_l equally match their associated covariance matrix \mathbf{A}_l (meaning the $\mathbf{r}_l^T(t_n)\mathbf{A}_l^{-1}\mathbf{r}_l(t_n)$ are equal), the respective likelihoods should be equal. However, the leading scalar term unjustly assigns higher probabilities to the filters where the determinant of the covariance matrix is smaller. Therefore, Maybeck and Stevens [63] propose removing that leading coefficient. Since each probability $p_j(t_n)$ is calculated by dividing by the sum of all the numerator expressions for each possible value of \mathbf{a}_l , each probability is maintained between zero and one, and the sum of the L probabilities equals one.

If several parameters are unknown, the number of Kalman filters running in parallel,

even for a coarse grid of possible parameter values, may be prohibitive. To circumvent this problem, Maybeck and Hentz [34, 61] propose applying a bank of Kalman filters for which the filter parameters are dynamically redeclared. Thus, the assumed parameter values, upon which the filters in the bank are based, move through the feasible parameter space. The small bank of Kalman filters searches for the best parameter estimates by redeclaring each filter's parameters \mathbf{a}_i . When the true parameter vector appears to be outside the current range of the parameters in the filter bank, one of two options is employed. The assumed filter parameters in the bank are either moved or their range expanded. As the bank converges to "good" estimates of the parameters, the range of parameters in the bank of filters may be contracted.

Many logical schemes for dynamically redeclaring the parameters in the bank of Kalman filters are possible. Maybeck and Hentz [61] test several rules and report that two rules, which position the center of the Kalman filter bank, worked well. The first rule maintains the probabilistic weighted average of the parameters $\hat{\mathbf{a}}(t_n)$ from Equation (39) near the center of the parameters in the bank, and the other rule positions the parameters \mathbf{a}_i with the highest probability $p_i(t_n)$ as the center. When the parameters in the filters are redeclared, the filters with new parameters are initialized with the probabilistic weighted average of the state estimate, Equation (37). In addition, all the filters with new parameters are assigned an equal share of the probability from the filters with discontinued parameters.

Maybeck and Hentz [61] also propose a contraction criterion and an expansion criterion for the range of parameters in the Kalman filter bank. They suggest contracting when a matrix norm of the conditional covariance matrix for $\hat{\mathbf{a}}(t_n)$, $\Sigma_{\mathbf{a}|\mathbf{Z}_n}$, falls below a certain threshold. Their expansion criterion is based on residual monitoring using the filter computed residual covariance matrix, $\mathbf{A}_i^{-1} = [\mathbf{H}_i(t_n)\mathbf{P}_i(t_n^-)\mathbf{H}_i^T(t_n) + \mathbf{R}_i(t_n)]^{-1}$. Specifically, their criterion is that if all the likelihood quotients

$$L_i(t_n) = \mathbf{r}_i^T(t_n)\mathbf{A}_i^{-1}\mathbf{r}_i(t_n)$$

are large and close in magnitude, the bank of filters is expanded. Both of these rules contract or expand the filters in all dimensions parameter space at the same time. More complicated approaches may contract or expand only in specific dimensions of the unknown parameter vector. An approach which expands and contracts in different directions would be useful to hone in on easily estimated parameters while continuing the coarse search for other parameters that are more difficult to estimate.

Harrison and Stevens [31, 32] apply an approach similar to MMAE to estimate time series data. Their technique uses a bank of four Kalman filters. Each filter is designed for a different condition; steady-state, transient value, slope change, or jump change. At each stage, the sixteen probabilities of transitioning from each filter to any filter are determined. To avoid a growing bank of filters, the probabilities and weighted state estimates are used to reduce back to the original four filters.

Like Akaike's method, the MMAE technique is another alternative approach to determine the system structure. MMAE may also be used as a method to estimate unknown system parameters. The MMAE technique, particularly with the use of the moving bank of filters, appears to require considerably less computation than Akaike's technique. However, many filters may be required to span different structural models each with unknown parameters.

3.4.3 Transfer Function Technique. Copper and Wood [16] relate the state-space representation to a transfer function model. The transfer function is a linear operator which relates the system inputs to outputs [11]. For example, the Autoregressive Moving Average (ARMA) time series models are equivalent to types of transfer functions which relate a normally-distributed white noise input to an output sequence. Given a stream of output data, the transfer function can be estimated either in the time or frequency domains. In the time domain, an ARMA model is fit, whereas in the frequency domain, the model is estimated by using a Fourier transformation. Lee [54:99-107] describes a methodology to identify the transition matrix Φ by transforming to canonical form and estimating the parameters with least squares.

3.4.4 A Priori Model Formulation. For specific applications, the state-space structure may be imposed on the simulation output sequence by *a priori* considerations. Under these circumstances, the dimension and structure of the problem are fixed, and the parameters can be estimated by least squares, Maximum Likelihood Estimation (MLE), Multiple Model Adaptive Estimation (MMAE) [58], or correlation techniques [9, 65, 66]. Harvey [33] presents the necessary likelihood functions for several structures which are commonly encountered in time series data.

3.5 Summary

The first major section of this chapter reviews the theory of stochastic processes. A discrete-time dynamics relationship, Equation (19), and a measurement model, Equation (20) are shown. Using these system equations, the next section develops the discrete-time, time-invariant, linear, Gaussian, Kalman filter equations. The final section in the chapter describes four potential techniques for determining and estimating the filter-design system equations for simulation output data. The next chapter applies these concepts to the specific problem of simulation steady-state identification.

IV. Single-Run Steady-State Identification: Residual Monitoring Approach

4.1 Introduction

In discrete-event simulation, a typical output analysis objective is to estimate the characteristics of the model outputs of interest, such as the mean or variance. One class of models is that of infinite-horizon simulations, in which our interest would be in estimating steady-state parameters. Although it is possible that these parameters may be cyclic, this research focuses on output sequences that attain a stationary steady-state probability distribution.

Typically, a simulation model is not initialized at steady-state conditions. The initial transient or startup problem arises from the fact that the model's output may initially pass through an unrepresentative phase as it approaches steady state. Including transient data in output analysis may bias steady-state estimates, as described in Chapter II. A generally accepted solution is to truncate the output sequence, thereby eliminating the initial data that may bias the steady-state estimates. While initial data truncation may increase the mean squared error of the point estimator [21, 104], deletion improves confidence interval coverage markedly [44]. This chapter proposes and tests an analytical approach to determine the appropriate initial-data truncation point.

Selection of the appropriate truncation point may be obfuscated by the difficulty that the transient, rather than ending at a particular point, may converge gradually to the steady-state distribution [101]. The effective end of the transient n_o is the point after which the initial conditions no longer have a significant effect on the estimated performance measures [15]. Deleting the data prior to n_o reduces the bias, but deleting data after n_o unnecessarily reduces the sample size and increases the variance of the estimates. The specific problem addressed by this chapter is estimating an appropriate truncation point \hat{n}_o for a univariate simulation output sequence $\{y_1, y_2, \dots, y_N\}$. For the confidence interval construction techniques designed for one long simulation run, such as batch means, this truncation point

identification algorithm, which also works on a single replication, is a useful complementary technique.

The background on related discrete-event simulation research is presented in Chapter II, and Chapter III describes the Kalman filter. This chapter is divided into four major sections. The proposed system models and a model estimation strategy is developed in the first two sections. The next section on the truncation point selection methodology discusses Kalman filter residual monitoring, and the following section presents the steps in the proposed algorithm. The results section reports on the Monte Carlo analyses of second-order autoregressive AR(2) data and first-order autoregressive-moving average ARMA(1,1) data. Although the residual monitoring approach does not perform as well as the MMAE approach in Chapter V, Section 4.2 provides the model formulation used in all of the subsequent methods.

4.2 Determination of the System Structure

The first step necessary in applying a Kalman filter is to determine appropriate system equations. Four techniques are presented in the last half of Chapter III. Akaike's minimum information criterion (MAICE), the Multiple Model Adaptive Estimation (MMAE), and transfer functions can be used to determine appropriate system equations from general classes of possible systems. In addition, the *a priori* method uses additional information to limit the possible structure of the dynamic and measurement equations. Limiting the possible structure and parameters significantly simplifies the system identification task. Kelton and Law [44] and Schruben [87] impose a structure on the simulation output and achieve good results. Therefore, using a similar structure for the form of the system equations, an *a priori* approach is developed. The first subsection presents an *a priori* model formulation. The associated Kalman filter equations and relationships are developed in the second subsection.

4.2.1 A Priori Model Formulation. In order to apply the Kalman filter to simulation output, an appropriate dynamics model, Equation (19) on page 49, and the associated

measurement model, Equation (20), must be determined. Since steady-state output is being modeled, time-invariant equations (33) and (34) on page 62, are chosen.

Abate and Whitt [1], Harrison and Williams [30], and Whitt [102] use regulated Brownian motion as the underlying model in analysis of queuing systems. A similar formulation was considered for the dynamics equation. However, this approach was rejected because of the nonlinear state transitions that would have to be taken into account in the Kalman filter.

Kelton and Law [44], as shown in Equation (13) on page 33, and Schruben [87], as shown in Equation (7) on page 28, suggest modeling the steady-state simulation output y_n as a constant mean μ_y plus noise $\eta(t_n)$:

$$y(t_n) = \mu_y + \eta(t_n) \text{ for observations } n = 1, 2, \dots, N \quad (41)$$

The term $\eta(t_n)$ is correlated noise with $E[\eta(t_n)] = 0$. These researchers attempt to account for the correlation in the noise term. Kelton and Law [44] use generalized least squares rather than ordinary least squares to account for the correlated noise term. Schruben [87], on the other hand, transforms the sequence so that the noise is modeled as a Brownian bridge. In both applications, the underlying model of the steady-state simulation output is a constant mean plus noise. In the Kalman filter approach, with the same model for the simulation output, the dynamics and measurement models must account for the correlation in the noise.

Using the same underlying model, the steady-state simulation output sequence $\{y(t_n)\}$ can be approximated by a steady-state mean μ_y plus a time-correlated noise $\eta(t_n)$. The sequence of time-correlated noise is approximated as the output of a linear stochastic dynamics system. The linear modeled noise terms $\tilde{\eta}(t_n)$ and their lagged values are the filter-design state vector $\mathbf{x}(t_n)$ in the system model. A similar concept is used by Fishman [20, 23] to construct confidence intervals for an estimate of the process mean. Fishman's approach fits an autoregressive time series model to the simulation output and uses the autoregressive coefficients to estimate the variance of the mean estimate. Fishman's technique applies

time-invariant autoregressive coefficients. Since steady-state output is being modeled, time-invariant system matrices seem reasonable.

Modeling the steady-state phase of simulation output by means of a linear time-invariant system has been questioned. In evaluating frequency domain experimentation, Sargent and Som [31] state that the input and output process of a modulated $M/M/1$ simulation does not constitute a time-invariant linear system. However, their analysis is confined to examining the simulation model from inputs to outputs. In their study, the input is the cosine of the average service time and the output is total time in the system. In contrast, the time-series approach assumes a linear time-invariant system between lagged output values. Thus, Sargent and Som's objections do not apply to this formulation.

The assumed constant autocorrelation relationship of a linear time-invariant system may be inappropriate. Using the simulation of an $M/M/1$ queue as an example, let the output of interest be either the waiting time for each entity or the total number in the queue. When the queue length is long, a strong autocorrelation in the simulation output seems logical. However, observations of either entity waiting times or queue lengths that are separated by a period of time when the server is idle are functions of nonoverlapping sequences of interarrival and service times. Both sequences are pseudorandom numbers, which are designed to be uncorrelated in time. Thus, when separated by a period with an idle server, no apparent autocorrelation may exist in the output for either waiting times or queue lengths. Furthermore, the degree of autocorrelation may increase with queue length. In spite of this objection, like Fishman's autoregressive time series approach [20, 23], this application assumes a linear time-invariant system between lagged simulation output values. The approximation ought to be sufficiently accurate for the purpose of steady-state identification.

An appropriate form and order of the noise model must be determined and estimated. Odoni and Roth [73] investigate the simulation state of queue length. They state that, for queuing systems during the transient, the sequence of the simulation states is dominated by a decaying exponential. Similarly, Conway [15] states that the transient components typically

decay geometrically. While Odoni and Roth's [73] and Conway's [15] conclusions apply to the transient phase of simulation output, they seem to indicate that the dominant autocorrelation in the output sequence can be approximated by relatively simple autoregressive models.

In time series analysis, a decaying exponential is modeled as a first-order autoregressive AR(1) process [11:57],

$$\tilde{\eta}(t_n) = \phi_1 \tilde{\eta}(t_{n-1}) + w_d(t_n) \quad (42)$$

where ϕ_1 is the first autoregressive coefficient and $\{w_d(t_n)\}$ is a sequence of white normally-distributed noise terms. A second-order autoregressive AR(2) process,

$$\tilde{\eta}(t_n) = \phi_1 \tilde{\eta}(t_{n-1}) + \phi_2 \tilde{\eta}(t_{n-2}) + w_d(t_n) \quad (43)$$

can model a mixture of damped exponentials and a sinusoidal within an exponentially decaying envelope [11:59-60].

Steudel and Wu [97] and Schriber and Andrews [85] report that periodic samples of $M/M/1$ queue length can be represented adequately as an AR(1). Charnes [12] reports that a vector of queue lengths from more complex simulation models can be modeled adequately as a first-order vector autoregressive, VAR(1), model.

In this research, the simulation output is modeled as a steady-state mean plus correlated noise. The true time-correlated noise $\eta(t_n)$ is approximated with autoregressive noise $\tilde{\eta}(t_n)$. The dynamics of the correlated noise is formulated either as an AR(1) with measurement noise model, which has a scalar filter-design state $x(t_n)$, or as an AR(2) with measurement noise model, which has a two-dimensional state vector $\mathbf{x}(t_n)$. Both approaches, the AR(1) with measurement noise and the AR(2) with measurement noise formulations, are developed and tested in this chapter. While the noise in simulations may have higher order effects, in many engineering applications, reduced order models are sufficient. In addition, the "lack of fit" of these simple models perhaps can be approximated as measurement noise. This a priori model of a steady-state mean plus autoregressive noise is imposed on discrete-event simulation output. The next subsection presents the corresponding Kalman

filter equations.

4.2.2 The Kalman Filter Equations. Autoregressive models can be written as discrete, linear, time-invariant, stochastic, dynamic systems with the appropriate choices of defining matrices. The filter-design system states $\mathbf{x}(t_n)$ are the autoregressive noises $\tilde{\eta}(t_n)$, and the state transition matrix Φ relates one state vector to the next. The noise dispersion matrix \mathbf{G}_d determines how the white normally-distributed scalar dynamic noise $w_d(t_n)$ affects the states. Assume that the noise correlation can be approximated with an autoregressive model. The dynamics model, Equation (33), and measurement model, Equation (34), matrices for an AR(1) with measurement noise formulation are

$$\mathbf{x}(t_n) = \tilde{\eta}(t_n), \quad \Phi = \phi_1, \quad \mathbf{G}_d = 1 \text{ and } \mathbf{H} = 1 \quad (44)$$

and for an AR(2) with measurement noise formulation are

$$\mathbf{x}(t_n) = \begin{bmatrix} \tilde{\eta}(t_n) \\ \tilde{\eta}(t_{n-1}) \end{bmatrix} \quad \Phi = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \quad \mathbf{G}_d = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (45)$$

With this formulation, $w_d(t_n)$, $Q_d(t_n)$, $z(t_n)$, $v(t_n)$, and $R(t_n)$ are scalars. The variances of the discrete, white, normally-distributed, zero-mean, dynamic noise $w_d(t_n)$ and measurement noise $v(t_n)$ are assumed to be constant for the steady-state simulation output. Thus, $Q_d(t_n) = Q_d$ and $R(t_n) = R$, respectively.

Using either of the two possibilities for \mathbf{H} , the modeled measurement $z(t_n)$ without the noise term $v(t_n)$ from Equation (34) is

$$\mathbf{H}\mathbf{x}(t_n) = \tilde{\eta}(t_n) \quad (46)$$

The actual measurements z_n of the correlated noises η_n are determined from the realization of simulation output y_n as follows from the random variable relationship expressed in

Equation (41):

$$z_n = y_n - \mu_y \quad (47)$$

Harvey [33:52-53] differentiates between the pure AR models without measurement noise and the AR models with measurement noise. Since the state is unobserved with measurement noise, Harvey calls the Autoregressive-Integrated-Moving Average (ARIMA) models with measurement noise Unobserved Components ARIMA (UCARIMA) models. A pure AR(1) process has theoretical autocovariances of the state $x(t_n)$ at lag τ of

$$\gamma_x(\tau) = \frac{\phi_1^\tau Q_d}{1 - \phi_1^2} \text{ for } \tau = 0, 1, 2, \dots$$

and has an autocorrelation function of

$$\rho_x(\tau) = \phi_1^\tau \text{ for } \tau = 0, 1, 2, \dots$$

In the ARMA case, one of the state vector elements $x(t_n)$ equals the observation $y(t_n)$ minus the steady-state mean μ_y . In contrast, the states in the UCARMA formulation are observed with measurement noise $v(t_n)$ with variance R . The variance of an UCAR(1) sequence includes the measurement noise variance,

$$\gamma(0) = \frac{\phi_1^\tau Q_d}{1 - \phi_1^2} + R$$

but the lagged autocovariances are the same as the AR(1) case since the measurement noise is uncorrelated in time. Therefore, the UCAR(1) autocorrelation function is

$$\rho_v(\tau) = \frac{\phi_1^\tau Q_d (1 - \phi_1^2)^{-1}}{Q_d (1 - \phi_1^2)^{-1} + R} \text{ for } \tau = 0, 1, 2, \dots$$

Inclusion of measurement noise requires that the variance R of the measurement noise $v(t_n)$ must also be estimated. Rather than using the Harvey's UCAR(1) and UCAR(2) notation, these formulations are referred to as the AR(1) with measurement noise model or the AR(2)

with measurement noise model.

Pure AR models do not include measurement noise $v(t_n)$, but they can be thought of as a degenerative form of the system model upon which Kalman filter is based. The degenerative form simply requires eliminating the measurement noise $v(t_n)$. Setting the variance of the measurement noise to zero, $R = 0$, removes the measurement noise from the system model since it already has a mean of zero. For the above autoregressive models, applying the Kalman filter equations with $R = 0$ results in the same estimates as applying a pure AR(1) or AR(2) model.

A simple first or second order autoregressive model may not be adequate since Fishman [20, 23] requires considerably more autoregressive terms in his applications to simulation output. Specifically, he requires four lagged values to model the output of an $M/M/1$ queue [20], seven lags for a simple airline reservation problem [23:256-257], and twenty-three lagged values for a cyclical washer problem [23:259]. However, these system equations provide two ways to account for "lack of fit" due to assuming the correlated noise $\eta(t_n)$ can be modeled adequately as a low-order autoregressive noise $\tilde{\eta}(t_n)$. Since discrete-event simulation output y_n is known exactly, the variance R of the measurement noise $v(t_n)$ may represent "lack of fit" between the true noise $\eta(t_n)$ and the assumed autoregressive noise $\tilde{\eta}(t_n)$. Second, reduced order models are applied in many engineering applications by simply increasing the variance Q_d of the dynamic driving noise $w_d(t_n)$ [57].

For the proposed model of the steady-state simulation output, the system matrices Φ , G_d , Q_d , H and R are assumed time-invariant. Therefore, the Kalman filter gain $K(t_n)$ and state-estimate variance matrices $P(t_n^-)$ and $P(t_n^+)$ attain steady-state values as discussed in Section 3.3.4. In this application, the steady-state matrices are used in all calculations. Since steady-state values are applied throughout, the time subscripts on the matrices K , P^- , P^+ are not necessary.

This a priori formulation can be implemented two different ways. The simulation output can be processed directly through this model of the Kalman filter as in most engineering

applications. Another approach revolves around the idea that the simulation output can first be smoothed by a moving average as in Welch's [101] technique. The moving average replaces each observation with the average of a predetermined number of past and future observations. This moving average removes some of the high frequency oscillations in the output. This smoothed data then can be processed through the Kalman filter to determine the truncation point. The approach taken in this chapter is to process the simulation observations directly.

In order to implement this formulation of the Kalman filter, the system parameters must be estimated. The AR(1) with measurement noise formulation has four unknown parameters: the steady-state mean μ_y , the variance Q_d of the dynamics driving noise $w_d(t_n)$, the variance R of the measurement noise $v(t_n)$, and the autoregressive coefficient ϕ_1 . The AR(2) with measurement noise formulation has an additional unknown autoregressive coefficient ϕ_2 to estimate. Their estimation is discussed in the next section.

4.3 System Parameter Estimation

The a priori formulation proposed in the previous section requires the system parameters to be estimated. These parameters are the observation mean μ_y , the autocorrelation coefficients ϕ_1 (and ϕ_2 for an AR(2) with measurement noise formulation), the dynamic noise variance Q_d , and the measurement noise variance R . Several estimation procedures, such as Multiple Model Adaptive Estimation (MMAE), Maximum Likelihood Estimation (MLE), least squares estimation, or correlation techniques [66], are possible approaches.

In the method of MMAE (See Section 3.4.2), a bank of Kalman filters, each with a different set of parameters, is run. The Maximum A Posterior (MAP) version of MMAE selects, as estimates, the parameters from the filter with the most probable residuals. The Bayesian version of MMAE uses the likelihood of each filter to calculate a weighted average for the parameter estimates, as shown in Equation (39).

Classical MLE selects as parameter estimates the values which maximize the joint probability of the actual measurements. Least squares estimation picks the parameter esti-

mates that minimize the sum of the squared residuals. Therefore, while MLE and MMAE are calculated based on the magnitude and assumed probability distribution of the Kalman filter residuals, least squares is based only on the size of the residuals. For a univariate sequence, least squares estimates are generally asymptotically equivalent to MLE [33:129].

The correlation technique proposed by Mehra [66] matches the theoretical autocorrelations for the assumed model with the sample autocorrelations. After solving for the mean and autocorrelation coefficients, the noise variances are calculated from the sample variance of the resulting residuals.

This section discusses the identifiability of the noise variances, Q_d and R . The effect of this indeterminacy between Q_d and R on the various estimation approaches is discussed. In order to eliminate the indeterminacy, the first term of the Kalman filter gain, $K_1 = k$, is estimated instead of the variances. Because of the lack of variance information, least squares estimation is selected. Using the estimated system parameters and the residuals, the noise variances, Q_d and R , the state estimate variances, P^- and P^+ , and the variances of the output and the mean estimator can be calculated.

4.3.1 Identifiability. A set of parameters is identifiable if no other set of parameters has the same joint density function [33:205].

Identifiability has an immediate bearing on estimation. If two structures have the same joint density function, the probability of generating a particular set of observations is the same for both structures. Thus there is no way of differentiating between them on the basis of the data. Furthermore, it often will be the case that attempts to estimate models which are not identifiable will run into practical difficulties [33:205].

Simultaneously estimating both the dynamics noise variance Q_d and the measurement noise variance R is an identifiability problem. As discussed on page 55 and in Section 3.3.4, the estimated state variances are not effected by the actual measurements. Further, the solution of Equation (32) gives the steady state solution for P^- . If the noise variances,

Q_d and R , are multiplied by a positive constant, P^- multiplied by the same constant is a solution of Equation (32). Using the scaled variances and calculating the Kalman filter gain with Equation (30) results in the original gain. Since the system matrices Φ, G_d, H, K along with the sequence of measurements determine the residuals, scaling all the variances by a positive constant does not change the resulting residuals. For scalar system, the Kalman filter gain is a function of the ratio of $\frac{Q_d}{R}$ [57:225], so any combination with the same $\frac{Q_d}{R}$ ratio generates the same residuals and has equal likelihood of being the correct model. Therefore, Q_d and R can not be estimated simultaneously.

4.3.2 Estimation Techniques. In spite of the confounding effect described in the previous subsection, an approach for making unique estimates of the system parameters is developed in this subsection. The estimation approach is based on "concentrating out" either Q_d or R . The effect of this approach on MLE, MMAE, and least squares estimation is discussed. Rather than "concentrating out" Q_d or R , estimation of the Kaman filter gain directly is examined and selected.

For a univariate sequence, a common approach, if both of the noise variances are unknown, is one of the variances should be "concentrated out" of the likelihood equation. Harvey [33:107,126] shows that by setting one variance as a scalar multiple of the other variance, one of the variances can be set arbitrarily and the other can be estimated using MLE without the problem of indeterminacy.

To understand Harvey's approach, let σ_*^2 be the unknown variance to be "concentrated out" of the likelihood function. Harvey begins by dividing all of the variances by σ_*^2 . Let $P_*^- = \frac{1}{\sigma_*^2} P^-$ and $P_*^+ = \frac{1}{\sigma_*^2} P^+$, and use $Q_{d*} = \frac{1}{\sigma_*^2} Q_d$ and $R_* = \frac{1}{\sigma_*^2} R$. The same Kalman filter variance propagation and update equations can be used, and the original Kalman filter gain K , Equation (30), is attained. Harvey shows that MLE can be applied with these scaled variances. The natural logarithm of the likelihood function for a multivariate sequence with time-invariant system matrices is Equation (35) on page 64. The version for a univariate

sequence with measurement noise with steady-state covariance matrices is

$$\ln(L) = -\frac{N}{2} \ln(2\pi) - \frac{N}{2} \ln(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + R) - \frac{\sum_{n=1}^N r_n^2}{2(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + R)}$$

where N is the number of observations

\mathbf{H} is the measurement matrix

\mathbf{P}^- is the covariance matrix of the state estimates prior to measurement update

R is the covariance matrix of the measurement noise

r_n is the residual for the n th observation

Substituting in the scaled variances,

$$\begin{aligned} \ln(L) &= -\frac{N}{2} \ln(2\pi) - \frac{N}{2} \ln(\sigma_*^2 \mathbf{H}\mathbf{P}_*^-\mathbf{H}^T + \sigma_*^2 R_*) - \frac{\sum_{n=1}^N r_n^2}{2(\sigma_*^2 \mathbf{H}\mathbf{P}_*^-\mathbf{H}^T + \sigma_*^2 R_*)} \\ &= -\frac{N}{2} \ln(2\pi) - \frac{N}{2} \ln(\sigma_*^2) - \frac{N}{2} \ln(\mathbf{H}\mathbf{P}_*^-\mathbf{H}^T + R_*) - \left(\frac{1}{2\sigma_*^2}\right) \frac{\sum_{n=1}^N r_n^2}{\mathbf{H}\mathbf{P}_*^-\mathbf{H}^T + R_*} \end{aligned}$$

Harvey differentiates with respect to σ_*^2 . Since the measurement sequence and the Kalman filter gain are unchanged by scaling with σ_*^2 , the residuals r_n are the same. In addition, the residual variance, $\mathbf{H}\mathbf{P}_*^-\mathbf{H}^T + R_*$ is not a function of σ_*^2 :

$$\frac{\partial \ln(L)}{\partial \sigma_*^2} = -\frac{N}{2\sigma_*^2} + \left(\frac{1}{2(\sigma_*^2)^2}\right) \frac{\sum_{n=1}^N r_n^2}{\mathbf{H}\mathbf{P}_*^-\mathbf{H}^T + R_*} = 0$$

Multiplying through by $2(\sigma_*^2)^2$ and solving,

$$\sigma_*^2 = \frac{\frac{1}{N} \sum_{n=1}^N r_n^2}{\mathbf{H}\mathbf{P}_*^-\mathbf{H}^T + R_*}$$

The second partial of $\ln(L)$ with respect to σ_*^2 is

$$\frac{\partial^2 \ln(L)}{\partial^2 \sigma_*^2} = \frac{N}{2(\sigma_*^2)^2} - \left(\frac{1}{(\sigma_*^2)^3}\right) \frac{\sum_{n=1}^N r_n^2}{\mathbf{H}\mathbf{P}_*^-\mathbf{H}^T + R_*} = \frac{-N}{2\sigma_*^4} - 1 < 0$$

Since σ_*^2 is a variance, it must be non-negative. Thus, the second partial of $\ln(L)$ is strictly positive, and setting the first partial equal to 0 yields the σ_*^2 which maximizes the natural

logarithm of the likelihood function.

Under the assumption of a linear system with known system matrices (Φ , G_d , Q_d , H and R), the residuals are zero-mean, white, random variables. An approximation of the residual sample variance equals $\frac{1}{N} \sum_{n=1}^N r_n^2$, and the filter-computed residual variance is $HP_*^-H^T + R_*$. Thus, σ_*^2 is approximately the ratio of the sample residual variance to the filter-computed residual variance.

Harvey [33:107,126] continues by substituting the ratio of variances for σ_*^2 in the likelihood equation:

$$\begin{aligned} \ln(L_c) &= -\frac{N}{2} \ln(2\pi) - \frac{N}{2} \ln \left(\frac{\frac{1}{N} \sum_{n=1}^N r_n^2}{HP_*^-H^T + R_*} \right) - \frac{N}{2} \ln(HP_*^-H^T + R_*) - \frac{N}{2} \\ &= -\frac{N}{2} (\ln(2\pi) + 1) - \frac{N}{2} \ln(HP_*^-H^T + R_*)^{-1} - \frac{N}{2} \ln \left(\frac{1}{N} \sum_{n=1}^N r_n^2 \right) - \frac{N}{2} \ln(HP_*^-H^T + R_*) \\ &= -\frac{N}{2} (\ln(2\pi) + 1) - \frac{N}{2} \ln \left(\frac{1}{N} \sum_{n=1}^N r_n^2 \right) \end{aligned}$$

This likelihood equation can be maximized to determine parameter estimates. Alternatively,

$$\ln(L_c^*) = N \ln \left(\frac{1}{N} \sum_{n=1}^N r_n^2 \right)$$

can be minimized. In contrast to the MLE parameters, the least squares estimates are determined by selecting parameters that minimize the sum of the residuals squared, $\sum_{n=1}^N r_n^2$. In general, MLE and least squares estimates are asymptotically equivalent estimates [33:129]. Since the natural logarithm is a monotonically increasing function, in this particular application, the MLE parameter estimates are equal to the least squares parameter estimates for fixed sample size N . However, in this application of trying to separate the transient and the steady-state, both estimation techniques, MLE and least squares, beg the question of how much of the data (the size of N) to use in estimating the steady-state parameters.

Harvey [33:126] recommends setting σ_*^2 equal to R or a diagonal element in Q_d . If $\sigma_*^2 = R$, then $R_* = 1.0$ and Q_{d*} becomes the dynamics-to-measurement noise ratio. Using the modified likelihood function, one less parameter is estimated, and the indeterminacy between Q_d and R should be eliminated. Either MLE or least squares estimation can be

used to estimate the remaining parameters.

The scaled variances can possibly be used in *normalized* Kalman filters in the MMAE technique. The conditional probability of the j th filter in the bank of K filters having the correct parameters is given by Equation (36) on page 69. For the case of scalar measurements, substituting in the normal distribution probability density function for the residuals with mean zero and variance of $\mathbf{HP}^{-}\mathbf{H}^T + R$ results in

$$\begin{aligned} p_j(t_n) &= \frac{f_{\mathbf{z}(t_n)|\mathbf{a}, \mathbf{Z}(t_{n-1})}(\mathbf{z}_n|\mathbf{a}_j, \mathbf{Z}_{n-1}) \cdot p_j(t_{n-1})}{\sum_{k=1}^K f_{\mathbf{z}(t_n)|\mathbf{a}, \mathbf{Z}(t_{n-1})}(\mathbf{z}_n|\mathbf{a}_k, \mathbf{Z}_{n-1}) \cdot p_k(t_{n-1})} \\ &= \frac{(2\pi(\mathbf{HP}_j^{-}\mathbf{H}^T + R_j))^{-1/2} \exp\left\{\frac{-r_j^2}{2(\mathbf{HP}_j^{-}\mathbf{H}^T + R_j)}\right\} \cdot p_j(t_{n-1})}{\sum_{k=1}^K (2\pi(\mathbf{HP}_k^{-}\mathbf{H}^T + R_k))^{-1/2} \exp\left\{\frac{-r_k^2}{2(\mathbf{HP}_k^{-}\mathbf{H}^T + R_k)}\right\} \cdot p_k(t_{n-1})} \\ &= \frac{(\mathbf{HP}_j^{-}\mathbf{H}^T + R_j)^{-1/2} \exp\left\{\frac{-r_j^2}{2(\mathbf{HP}_j^{-}\mathbf{H}^T + R_j)}\right\} \cdot p_j(t_{n-1})}{\sum_{k=1}^K (\mathbf{HP}_k^{-}\mathbf{H}^T + R_k)^{-1/2} \exp\left\{\frac{-r_k^2}{2(\mathbf{HP}_k^{-}\mathbf{H}^T + R_k)}\right\} \cdot p_k(t_{n-1})} \end{aligned}$$

Substituting in the scaled variances and σ_{*j}^2 for each filter into the MMAE probability calculations,

$$p_{*j}(t_n) = \frac{\sigma_{*j}^2 (\mathbf{HP}_{*j}^{-}\mathbf{H}^T + R_{*j})^{-1/2} \exp\left\{\frac{-r_j^2}{2\sigma_{*j}^2 (\mathbf{HP}_{*j}^{-}\mathbf{H}^T + R_{*j})}\right\} \cdot p_{*j}(t_{n-1})}{\sum_{k=1}^K \sigma_{*k}^2 (\mathbf{HP}_{*k}^{-}\mathbf{H}^T + R_{*k})^{-1/2} \exp\left\{\frac{-r_k^2}{2\sigma_{*k}^2 (\mathbf{HP}_{*k}^{-}\mathbf{H}^T + R_{*k})}\right\} \cdot p_{*k}(t_{n-1})}$$

Using Harvey's suggestion of setting $\sigma_{*j}^2 = R$, so $R_{*j} = 1.0$, results in the following probability calculations:

$$p_{*j}(t_n) = \frac{\sigma_{*j}^2 (\mathbf{HP}_{*j}^{-}\mathbf{H}^T + 1.0)^{-1/2} \exp\left\{\frac{-r_j^2}{2\sigma_{*j}^2 (\mathbf{HP}_{*j}^{-}\mathbf{H}^T + 1.0)}\right\} \cdot p_{*j}(t_{n-1})}{\sum_{k=1}^K \sigma_{*k}^2 (\mathbf{HP}_{*k}^{-}\mathbf{H}^T + 1.0)^{-1/2} \exp\left\{\frac{-r_k^2}{2\sigma_{*k}^2 (\mathbf{HP}_{*k}^{-}\mathbf{H}^T + 1.0)}\right\} \cdot p_{*k}(t_{n-1})}$$

The σ_{*j}^2 do not cancel and are unknown. The approach of assuming the σ_{*j}^2 cancel and setting an arbitrary value for each R_{*j} may result in a nonlinear change in the MMAE probability calculations.

In a test of MMAE with the scaled variances, the ϕ_1 and Q_d were varied at discrete levels for each of K filters. The MMAE probability calculations used

$$p_{*j}(t_n) = \frac{(\mathbf{HP}_{*j}^- \mathbf{H}^T + R_{*j})^{-1/2} \exp\left\{\frac{-r_j^2}{2(\mathbf{HP}_{*j}^- \mathbf{H}^T + R_{*j})}\right\} \cdot p_{*j}(t_{n-1})}{\sum_{k=1}^K (\mathbf{HP}_{*k}^- \mathbf{H}^T + R_{*k})^{-1/2} \exp\left\{\frac{-r_k^2}{2(\mathbf{HP}_{*k}^- \mathbf{H}^T + R_{*k})}\right\} \cdot p_{*k}(t_{n-1})}$$

Of the three discrete values (0.1, 0.5, and 0.9) for ϕ_1 , the center value of 0.5 was used to generate the AR(1) data. After processing 1,000 observations through the Kalman filters, the marginal probabilities for ϕ_1 were examined. For each run with different values of R_{*j} , the marginal probability for one of the discrete values of ϕ_1 was over 98 percent. This estimate of ϕ_1 constitutes the Maximum A Posteriori (MAP) estimate. However, different values of R_{*j} resulted in different MAP estimates of ϕ_1 . With $R_{*j} = 1.0$, ϕ_1 was overestimated as 0.9. On the other hand, when $R_{*j} = 0.1$, ϕ_1 was underestimated as 0.1. In neither case did MMAE closely estimate the correct ϕ_1 value of 0.5. In addition, although the actual filter residuals were almost identical for each value of R_{*j} , the filter-computed residual variance, $\mathbf{HP}_{*j}^- \mathbf{H}^T + R_{*j}$, varied significantly. In order to draw reliable statistical inferences about the residuals, the filter-computed residual variance should be approximately equal to the actual residual variance. The choice of this example was unfortunate since the correct value for R for pure AR(1) data without measurement noise is zero. Therefore, the MMAE was attempting to select a very large Kalman filter gain proportional to the ratio of $\frac{Q_d}{R}$. However, the approach of arbitrarily fixing Q_d or R and using MMAE to estimate the other parameters does not appear to be a reliable estimation strategy. Because of potential estimation problems and unreliable filter-computed variance estimates, MMAE is not applied in the algorithm developed in this chapter.

In an extension of Harvey's approach, the original (unstarred) parameters can be estimated after determining estimates of the starred (*) parameters. These calculations require estimating σ_e^2 with the mean squared residual, $\frac{1}{N} \sum_{n=1}^N r_n^2$, and the filter-computed residual variance, $\mathbf{HP}_{*j}^- \mathbf{H}^T + R_{*j}$. The advantage of this method is that the steady-state filter variance would be *tuned* [57:337-338] to the realized residual variance. Tuning prevents di-

vergence caused by the filter relying too heavily on its internal dynamic model, and tuning also prevents chasing noise in the data. Tuning the filter also ensures that the probability calculations for the residuals are reasonable.

Since Harvey's suggested approach scales the variances and MMAE probably should not be applied, a simpler formulation is to let $\sigma_*^2 = \mathbf{H}\mathbf{P}^-\mathbf{H}^T + R$, the filter-computed residual variance. This eliminates the need to calculate the variances Q_d , R , \mathbf{P}^- or \mathbf{P}^+ . The unknown parameters become the steady-state mean μ_y , a scalar k related to the Kalman filter gain \mathbf{K} , and the autoregressive coefficient(s) ϕ_1 (and ϕ_2) in Φ . The observations can be processed through Kalman filters with estimates of these unknown parameters without corresponding estimates for Q_d , R , \mathbf{P}^- or \mathbf{P}^+ . With only an estimate of the Kalman filter gain, but no variance estimates, MMAE and MLE can not be applied, however, least squares estimation remains feasible.

In this chapter, the least squares estimates for the steady-state mean μ_y , the autoregressive coefficients ϕ_1 and ϕ_2 , and the scalar k are used. These parameters fix the state transition matrix Φ and the Kalman filter gain \mathbf{K} . Based on these parameter estimates and the resulting residual sequence, the variances can be estimated as shown in the next subsection.

4.3.3 Variance Estimates. The variance relationships for both the AR(1) with measurement noise and the AR(2) with measurement noise models are developed. Using estimates for the output mean, autoregressive coefficients, and the first element of the Kalman filter gain, the simulation output sequence is processed through the Kalman filter. The resulting Kalman filter residuals along with the parameter estimates are used to calculate variance estimates. If desired, the unknown variances of R and Q_d , and thus \mathbf{P}^- and \mathbf{P}^+ , are calculated using the mean squared residuals. Further, based on the parameter estimates and the estimates of Q_d and R , the simulation output variance and the mean estimator variance are determined.

In the AR(1) with measurement noise formulation, $\mathbf{H} = \mathbf{G}_d = 1$, $\mathbf{K} = k$ and $\Phi = \phi_1$.

Using the estimates for μ_y , ϕ_1 and k , the data is processed through the Kalman filter. The approximate residual sample variance is set equal to the filter-computed residual variance:

$$\frac{1}{N} \sum_{n=1}^N r_n^2 = \mathbf{H}P^-\mathbf{H}^T + R = P^- + R$$

Using the Equation (30) for the Kalman gain,

$$k = \frac{P^-}{P^- + R} \quad (48)$$

After multiplying and substituting, this relationship results in

$$P^- = k(P^- + R) = k \left(\frac{1}{N} \sum_{n=1}^N r_n^2 \right)$$

By substituting into the residual variance equation, R is determined:

$$R = (1 - k) \left(\frac{1}{N} \sum_{n=1}^N r_n^2 \right)$$

Furthermore, the Kalman filter equations with the estimate for P^- are used to solve for P^+ and Q_d . From Equation (31),

$$P^+ = P^- - kP^-$$

Using Equation (29),

$$Q_d = P^- - \phi_1^2 P^+$$

In addition, since $k = \frac{P^-}{P^- + R}$ and both P^- and R are variances, the scalar k is bounded between zero and one. For an AR(1) with measurement noise formulation, the noise variances and the state estimate variances are determined from the parameter estimates and the mean of the squared residuals with these relationships.

For the AR(2) with measurement noise model, the Kalman filter gain is a vector with two elements. After setting the first element equal to the scalar k , the second element must

be solved in terms of the estimated parameters. Using the matrices from Equation (45) in Equation (30),

$$\mathbf{K} = \begin{bmatrix} P_{11}^- \\ P_{12}^- \end{bmatrix} \left(\frac{1}{P_{11}^- + R} \right) = \begin{bmatrix} \frac{P_{11}^-}{P_{11}^- + R} \\ \frac{P_{12}^-}{P_{11}^- + R} \end{bmatrix} \quad (49)$$

The first element of the Kalman filter gain matrix is set equal to the scalar k , so $k = \frac{P_{11}^-}{P_{11}^- + R}$. As in the AR(1) case, the parameter k is bounded; $0 \leq k \leq 1$. However, to determine the second element in the Kalman filter gain matrix, $\frac{P_{12}^-}{P_{11}^- + R}$ must be solved in terms of the estimated parameters ($\hat{\mu}_y$, \hat{k} , $\hat{\phi}_1$, and $\hat{\phi}_2$). P_{12}^- is determined using the Kalman filter equations along with the symmetry of covariance matrices. From the covariance update, Equation (31),

$$\begin{aligned} \mathbf{P}^+ &= \mathbf{P}^- - \mathbf{KHP}^- \\ &= \mathbf{P}^- - \mathbf{P}^- \mathbf{H}^T [\mathbf{HP}^- \mathbf{H}^T + \mathbf{R}]^{-1} \mathbf{HP}^- \\ &= \mathbf{P}^- - \left(\frac{1}{P_{11}^- + R} \right) \mathbf{P}^- \mathbf{H}^T \mathbf{HP}^- \\ &= \mathbf{P}^- - \left(\frac{1}{P_{11}^- + R} \right) \begin{bmatrix} (P_{11}^-)^2 & P_{11}^- P_{12}^- \\ P_{11}^- P_{12}^- & (P_{12}^-)^2 \end{bmatrix} \\ &= \begin{bmatrix} P_{11}^- - \frac{(P_{11}^-)^2}{P_{11}^- + R} & P_{12}^- - \frac{P_{11}^- P_{12}^-}{P_{11}^- + R} \\ P_{12}^- - \frac{P_{11}^- P_{12}^-}{P_{11}^- + R} & P_{22}^- - \frac{(P_{12}^-)^2}{P_{11}^- + R} \end{bmatrix} \end{aligned}$$

From the covariance matrix propagation equation, Equation (29),

$$\begin{aligned} \mathbf{P}^- &= \Phi \mathbf{P}^+ \Phi^T + \mathbf{G}_d \mathbf{Q}_d \mathbf{G}_d^T \\ &= \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} P_{11}^- - \frac{(P_{11}^-)^2}{P_{11}^- + R} & P_{12}^- - \frac{P_{11}^- P_{12}^-}{P_{11}^- + R} \\ P_{12}^- - \frac{P_{11}^- P_{12}^-}{P_{11}^- + R} & P_{22}^- - \frac{(P_{12}^-)^2}{P_{11}^- + R} \end{bmatrix} \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} + \begin{bmatrix} Q_d & 0 \\ 0 & 0 \end{bmatrix} \quad (50) \end{aligned}$$

After multiplying,

$$P_{12}^- = \phi_1 P_{11}^- - \frac{\phi_1 (P_{11}^-)^2}{P_{11}^- + R} + \phi_2 P_{12}^- - \frac{\phi_2 P_{11}^- P_{12}^-}{P_{11}^- + R}$$

From the equation for the first element of the Kalman filter gain matrix, $P_{11}^- = k(P_{11}^- + R)$. Substituting for P_{11}^- in the numerator terms,

$$P_{12}^- = \phi_1 k(P_{11}^- + R) - \phi_1 k^2(P_{11}^- + R) + \phi_2 P_{12}^- - \phi_2 k P_{12}^-$$

and solving for P_{12}^-

$$P_{12}^-(1 - \phi_2 + \phi_2 k) = (\phi_1 k - \phi_1 k^2)(P_{11}^- + R)$$

$$P_{12}^- = \frac{(\phi_1 k - \phi_1 k^2)}{(1 - \phi_2 + \phi_2 k)}(P_{11}^- + R)$$

Further,

$$\frac{P_{12}^-}{P_{11}^- + R} = \frac{\phi_1 k - \phi_1 k^2}{1 - \phi_2 + \phi_2 k}$$

Since this is the second element of the Kalman filter gain, the complete gain matrix is

$$\mathbf{K} = \begin{bmatrix} k \\ \frac{\phi_1 k - \phi_1 k^2}{1 - \phi_2 + \phi_2 k} \end{bmatrix} \quad (51)$$

With these relationships, the least squares estimates are found. After the least squares estimates of $\hat{\mu}_y$, \hat{k} , $\hat{\phi}_1$, and $\hat{\phi}_2$ are determined, using the relationships above, unique estimates of \mathbf{P}^- , \mathbf{P}^+ , \hat{Q}_d , and \hat{R} can be calculated, if desired. From the estimation of $\hat{\mu}$, \hat{k} , $\hat{\phi}_1$, and $\hat{\phi}_2$, a residual sequence exists. Theoretically, the residuals are normally distributed with mean zero and variance $\mathbf{H}\mathbf{P}^-\mathbf{H}^T + R$. An approximation of the residual sample variance is $\frac{1}{N} \sum_{n=1}^N r_n^2$. Equating these variances results in

$$\frac{1}{N} \sum_{n=1}^N r_n^2 = \mathbf{H}\mathbf{P}^-\mathbf{H}^T + R = P_{11}^- + R \quad (52)$$

From Equations (49), (51) and (52), P_{11}^- and P_{12}^- are determined. Again using Equation (52), \hat{R} is estimated. Finally, using Equation (50) and the fact that $P_{22}^- = P_{11}^+$, \hat{Q}_d is calculated.

In many simulation applications, an estimate of the variance of the process is useful. From fitting these Kalman filter equations, an estimate of the variance $\hat{\sigma}_y^2$ of the simulation

output y_n and an estimate of the variance $\hat{\sigma}_{\hat{\mu}_y}^2$ of the simulation output mean estimator $\hat{\mu}_y$ can be determined.

The variance of the output $\hat{\sigma}_y^2$ is derived using Equations (34), (46) and (47):

$$\begin{aligned} y_n &= \mu_y + z_n \\ &= \mu_y + \mathbf{H}\mathbf{x}(t_n) + v(t_n) \\ &= \mu_y + \tilde{\eta}(t_n) + v(t_n) \end{aligned} \quad (53)$$

The measurement noise $v(t_n)$ is assumed independent of the state estimates, which are the autoregressive noises $\tilde{\eta}(t_n)$. The independence along with the facts that $E\{v^2(t_n)\} = R$ and that $E\{\tilde{\eta}^2(t_n)\} = \sigma_{\tilde{\eta}}^2$ are used in the derivations.

For the AR(1) with measurement noise formulation, the $\tilde{\eta}(t_n)$ are modeled as a zero mean AR(1) process, as shown in Equation (42). Thus, $\sigma_{\tilde{\eta}}^2 = E[\tilde{\eta}^2(t_n)] = \left(\frac{Q_d}{1-\phi_1^2}\right)$ [11:58] of

$$\begin{aligned} \sigma_y^2 &= E\{[\mu_y + \tilde{\eta}(t_n) + v(t_n)]^2\} - \mu_y^2 \\ &= E\{\tilde{\eta}^2(t_n)\} + E\{v^2(t_n)\} \\ &= \left(\frac{Q_d}{1-\phi_1^2}\right) + R \end{aligned}$$

If the simulation output y_n is from an AR(1) process, no "lack of fit" is necessary, and $R = 0$. As expected, with $R = 0$, the variance is the same as for an AR(1) process.

For the AR(2) with measurement noise formulation, the correlated noise terms $\tilde{\eta}(t_n)$ are distributed as a pure AR(2) process, Equation (43), with process variance [11:62]:

$$\sigma_{\tilde{\eta}}^2 = E\{\tilde{\eta}^2(t_n)\} = \left(\frac{1-\phi_2}{1+\phi_2}\right) \left(\frac{Q_d}{(1-\phi_2)^2 - \phi_1^2}\right) \quad (54)$$

Since the result $E\{y_n\} = \mu_y$ still holds, only the last part of the derivation that uses $E\{y_n^2\}$ changes:

$$\begin{aligned} \sigma_y^2 &= E\{\tilde{\eta}^2(t_n)\} + R \\ &= \left(\frac{1-\phi_2}{1+\phi_2}\right) \left(\frac{Q_d}{(1-\phi_2)^2 - \phi_1^2}\right) + R \end{aligned} \quad (55)$$

Once again, if the actual process is an AR(2), then $R = 0$ and the variance of $y(t_n)$ degenerates to the pure AR(2) process variance.

In discrete-event simulations, the objective often is to calculate an estimate of the mean output $\hat{\mu}_y$. The variance of this estimate is necessary to form a confidence interval for the mean.

By independence of the state estimates $\mathbf{x}(t_n) = \tilde{\eta}(t_n)$ and the measurement noise $v(t_n)$, the variance of their sum is the sum of their variances.

$$\begin{aligned}\sigma_{\hat{\mu}_y}^2 &= \text{Var}\left\{\frac{1}{N} \sum_{n=1}^N y_n\right\} \\ &= \text{Var}\left\{\frac{1}{N} \sum_{n=1}^N \tilde{\eta}(t_n)\right\} + \frac{1}{N^2} \sum_{n=1}^N \text{Var}\{v(t_n)\} \\ &= \text{Var}\left\{\frac{1}{N} \sum_{n=1}^N \tilde{\eta}(t_n)\right\} + \frac{R}{N}\end{aligned}$$

Substituting Fishman's [23:247-252] approximation, Equation (11) with $p = 2$, for the variance of the mean estimator of an AR(2) sequence,

$$\sigma_{\hat{\mu}_y}^2 \approx \frac{Q_d}{N(1 - \phi_1 - \phi_2)^2} + \frac{R}{N} \quad (56)$$

with degrees of freedom d from Equation (12) minus 1 for the estimation of R ,

$$d = \frac{N(1 - \hat{\phi}_1 - \hat{\phi}_2)}{4 + 4\hat{\phi}_2} - 1 \quad (57)$$

Substituting in the parameter estimates results in an estimate of the output variance $\hat{\sigma}_y^2$. This variance estimate is used for constructing confidence intervals on the mean estimate [37, 39].

The variance estimates are derived from the estimate of the scalar k . Empirical tests shown in Table 53 on page 168 indicate that the scalar k is overestimated. The bias in estimating k also biases the variance estimates \hat{Q}_d and \hat{R} . Furthermore, estimates based on \hat{Q}_d and \hat{R} , such as $\hat{\sigma}_y$ and $\hat{\sigma}_{\hat{\mu}_y}$, may also be biased.

4.3.4 Application of Estimation Procedure. Two major issues affect the application of the estimation procedure developed in the last subsection. The first issue is in which direction should the data be processed, forward or backwards. In this chapter, the data is processed backwards beginning at the end of the simulation output, however, in subsequent chapters the data is processed forward from the beginning to end. The second issue is how many of the observations should be used to estimate the steady-state system model. All of the data is not used since the transient data may bias the parameter estimates necessary for the algorithm to detect the transient. A bank of filters is employed to simultaneously evaluate the feasible range of parameters until adequate parameters are accepted. The discretization for the parameters upon which the filters are based is also discussed. Various stopping rules are considered to determine when the estimates from one of the filters in the bank should be accepted as the steady-state system model.

Processing the data from the end of the simulation output toward the beginning minimizes the chance of biasing the estimated steady-state model with transient observations. Another advantage of estimating the steady-state model by processing the data from the end to the beginning is that after the model is estimated, the model's residuals may be tracked as a potential indication of the onset of significant transient effects.

In applying the least squares estimation technique, the decision must be made at which observation to stop the system parameter estimation. Processing the output backward from the simulation termination, in the same fashion as Kelton and Law [44] propose, the steady-state output is encountered before the transient output. While more steady-state observations should improve the confidence in the estimated parameters, including transient output may bias the system parameter estimates. But at this stage in the algorithm, no estimate of where the transient phase effectively ends is available. Therefore, at a point sufficiently early in the steady-state observations to prevent including any transient data, the parameter estimation procedure should be stopped.

The least squares estimates are generally found by applying Kalman filters with various parameters to a fixed sample. A search is conducted until the parameters which minimize

the sum of squared residuals is determined. In this application, the number of observations to include in the sample for the estimation algorithm is not obvious. The decision of how many observations to include in the parameter estimation phase is solved partially by using a bank of filters similar to the MMAE approach. The bank of filters are set up in a grid covering the feasible and stable parameter space. The bank of filters processes the observations backwards, beginning with the last observation. Using the bank of filters, various combinations of parameters are evaluated after including each additional observation to the estimation sample. As observations are processed, the cumulative sum of residuals squared is maintained for each filter. After sufficient observations are processed, one filter may consistently minimize the cumulative sum of squared residuals. This filter's parameters are selected as the least squares estimates. As soon as the best combinations of parameter estimates is apparent, no more data points are included in the estimation sample. Rather than using a predetermined sample size, this approach applies an incrementally increasing sample size until the estimates are determined. With the appropriate stopping rule, this bank of filters approach avoids having to determine which observations constitute steady state before beginning the estimation.

While optimized discretization schemes exist [94, 61], these schemes require a "truth model" which is not available in these applications. An equally spaced discretization is employed. The grid of filters covers the parameter space with a filter running at each combination of the discrete parameters. For each filter in the bank, the sum of its residuals squared is maintained. After processing an observation, the least square estimates to that point are the parameters of the filter with the minimum sum of squared residuals.

For initial tests, the Kalman filter formulation is an AR(1) with measurement noise model, and the test data sequence is generated with a pure AR(1) model. The filter bank includes filters with the scalar k discretized from 0.0 to 1.0 at every 0.1, ϕ_1 discretized from -1.0 to 1.0 at every 0.1. The true mean is used in the filters. After processing 1000 observations, the minimum is always very close to the parameters used to generate the data. However, the residual sum of squares is relatively insensitive to changes in the scalar k . In

Table 6. Estimation Stopping Criteria

Residual Properties	Hypothesis Test
Zero Mean	Mean Hypothesis Test
Known Covariance	χ^2 Variance Test
Whiteness	Potmanteau Test
(Uncorrelated in Time)	Durbin-Watson Statistic
Normally Distributed	Goodness of Fit Test

addition, most of the neighboring filters have residual sum of squares just a small percentage ($< 2\%$) larger than the minimum. In other words, the residual sum of squares appears to be a relatively flat surface. Therefore, parameter estimates that differ by less than 0.1 may not be distinguishable without an excessive number of observations. As a result of this conclusion, the parameters are estimated only to within 0.1 in this chapter. This coarse discretization enables the bank of Kalman filters to span the discretized feasible parameter space completely.

After determining the discretization of the unknown parameters, a criterion to accept the estimates from the filter with the minimum cumulative sum of squared residuals is necessary. One approach for determining a stopping rule is periodically to estimate the parameters and to test the fit of the model. Mehra and Peschon [64] present three types of statistical hypothesis tests for the residuals: test of zero mean, test of covariance, and test of whiteness. The assumption of normally distributed residuals can also be tested using a goodness of fit test. These and other possible techniques to determine the estimation stopping rule, shown in Table 6, are discussed. Because of the difficulty of implementing sequential statistical hypothesis tests, a heuristic approach is selected.

Two variants of the test for zero mean are examined. The hypothesis test to determine if the residuals are statistically zero-mean uses the sample mean, $\bar{r} = \frac{1}{N} \sum_{n=1}^N r_n$. The sample mean is distributed normally with zero mean and variance of $\frac{1}{N}[\mathbf{HP}^{-1}\mathbf{H}^T + R]$. Therefore,

the null hypothesis of zero mean is rejected whenever

$$|\bar{r}| > \frac{1.96[\mathbf{HP}^{-1}\mathbf{H}^T + R]}{\sqrt{N}}$$

at the 5 percent confidence level.

While the previous test is based on the filter-computed residual variance $\mathbf{HP}^{-1}\mathbf{H}^T + R$, Mehra and Peschon [64] recommend using the sample variance. Let the standardized residuals be $\tilde{r}_n = (\mathbf{HP}^{-1}\mathbf{H}^T + R)^{-\frac{1}{2}}r_n$, and the sample variance equal $\hat{\sigma}_{\tilde{r}}^2 = (N-1)^{-1} \sum_{n=1}^N (\tilde{r}_n - \bar{\tilde{r}})^2$. Mehra and Peschon suggest a T^2 test with the following statistic:

$$T^2 = N\hat{\sigma}_{\tilde{r}}^{-2}\bar{\tilde{r}}^2$$

which is the “uniformly most powerful among all the tests for zero mean which are invariant with respect to scaling (or covariance)” [64].

The χ^2 hypothesis test evaluates whether the residual variance equals the filter-computed variance. Since the estimation methodology forces these two variances to be equal, this test is not appropriate to determine if the parameter estimates are sufficient. However, this test is discussed later as a viable test for the onset of the transient data.

Whiteness is defined as independence in time. The test of whiteness is based on estimating the sample autocorrelations c_i at lag i with

$$\hat{c}_i = \frac{\sum_{n=i+1}^N (r_n - \bar{r})(r_{n-i} - \bar{r})}{\sum_{n=1}^N (r_n - \bar{r})^2}$$

These sample autocorrelations can be compared with their standard errors, which are approximated usually by $N^{-\frac{1}{2}}$. The difficulty with this approach is that for a few lags i , the true standard errors may be much smaller than the approximation [2:261].

Another test of the sample autocorrelations is the modified Portmanteau test for an ARMA(p, q) time series model [2:263]. In this test with N observations and K_Q sample

autocorrelations, the test statistic Q^* is given by

$$Q^* = N(N+2) \sum_{i=1}^{K_Q} \frac{\hat{c}_i}{N-i}$$

The statistic Q^* is χ^2 distributed with $K_Q - p - q$ degrees of freedom where $p + q$ is the total number of parameters in the model.

While the Portmanteau test evaluates K_Q autocorrelations, the Durbin-Watson statistic only tests if the first autocorrelation coefficient is near zero. The Durbin-Watson statistic,

$$D = \frac{\sum_{n=2}^N (r_n - r_{n-1})^2}{\sum_{n=1}^N r_n^2}$$

is approximately equal to $2(1 - \rho_r(1))$, where $\rho_r(1)$ is the first order autocorrelation coefficient of the residuals. Since theoretically $\rho_r(1)$ should be zero, D should be close to 2. In addition, a good heuristic test for whiteness is to adjust for the mean and then count the number of zero crossing.

A different approach is to apply a goodness-of-fit test to evaluate the cumulative distribution of the residuals. Several goodness-of-fit tests [96], such as the Kolmogorov-Smirnov test, can be used to test whether the residuals are distributed normally with a mean of zero and a variance equal to the filter-computed variance. The residuals for the test can be obtained by processing a different set of observations than the set of observations used to estimate the parameters.

If the residuals from the estimated model failed any of the hypothesis tests, more observations can be included in the the least squares estimation. However, a difficulty exists with repeatedly applying a statistical hypothesis test to determine if the estimated model is adequate. Due to its statistical nature, an incorrect model may eventually be accepted erroneously. The probability of failing to reject an inadequate model depends on the power of the hypothesis test, but all statistical tests have some probability of accepting an incorrect model. Therefore, for any given model, the test can be applied repeatedly to

determine if the acceptance rate is commensurate with the confidence level of the statistical test. However, repeatedly applying a statistical test and monitoring the rate of acceptance is not implemented easily. Therefore, sequential hypothesis tests are not applied.

Since applying a statistical hypothesis test to determine whether the parameter estimates are adequate is cumbersome, a simple heuristic approach is developed. The hypothesis tests are examined for values that can be easily monitored and give a relatively good indication of the quality of the parameter estimates. The assumption of zero-mean residuals can be tested by monitoring the cumulative sum of the residuals. If the residuals are zero mean and normally distributed, as assumed in the proposed system model, the cumulative sum of the residuals should fluctuate stochastically around zero [70:282]. Therefore, the parameters of the filter with the smallest sum of residuals squared are accepted only if the absolute value of the cumulative sum of residuals is within a small tolerance of zero.

Similarly, the Durbin-Watson statistic can be updated easily with each new observation. Thus, as an indication that the residuals are white, the parameter estimates are accepted only if the associated Durbin-Watson statistic is above 1.9 and below 2.1.

These stopping rules indicate that the residuals appear to follow the assumptions of being zero-mean and white. In addition, these rules are monitored easily as the observations are processed, and they do not require calculating the Kalman filter variance matrices. Therefore, the parameters of the Kalman filter with the minimum sum of squared residuals are tested after each additional observation is processed. When both tests are passed simultaneously, that filter's parameters are selected as the least squares estimates. The least squares estimation with these stopping rules completes the steady-state system identification. After several tests runs, the Durbin-Watson criterion was discarded. If the model used to generate the data does not match the Kalman filter formulation, the residuals continually fail the Durbin-Watson criterion.

The selected stopping rule is based on monitoring the cumulative sum of the residuals. The cumulative sum should be normally distributed with mean zero and variance propor-

tional to the sample size. The simulation output is processed backwards from the simulation end to the beginning. After processing $\{y_N, y_{N-1}, \dots, y_{N-J}\}$ for $J = 0, 1, 2, \dots$, the number of generated residuals is $J + 1$. The parameters of the filter with the smallest sum of residuals squared are accepted only if the absolute value of the cumulative sum of residuals is below a small tolerance times $\sqrt{J + 1}$:

$$\left| \sum_{j=0}^J r_{N-j} \right| \leq 0.1 \sqrt{J + 1} \quad (58)$$

In summary, a bank of filters, each with different parameter estimates, processes the data from the end of the simulation output backwards toward the beginning. From the bank of filters, the parameters of the Kalman filter with the minimum sum of squared residuals are tested after processing each additional observation. When the cumulative sum is less than a small tolerance, those filter's parameters are selected as the least squares estimates.

In this section, a procedure to estimate the system parameters is developed. The procedure processes the data from the end of the simulation output until the adequate parameter estimates are obtained. Based on this estimated steady-state model, the Kalman filter can continue processing the observations backwards to detect when transient data is encountered. The next section develops the algorithm for detecting transient data and selecting an appropriate truncation point.

4.4 *Truncation Point Selection*

Once the steady-state system identification is accomplished, determination of the point where the transient phase effectively ends may be addressed. The concept is to truncate the initial transient data in order to obtain a better estimate of the steady-state simulation output mean. Potentially, the appropriate truncation point may be identified in several different ways. This chapter tests an approach based on monitoring the Kalman filter residuals. When the simulation output is processed through the estimated Kalman filter from the end to the beginning, the transient effects may be apparent by the onset of a high occurrence of

Table 7. Truncation Point Selection Criteria

Residual Properties	Hypothesis Test
Zero Mean	Mean Hypothesis Test
Known Covariance	χ^2 Variance Test
Whiteness (Uncorrelated in Time)	Durbin-Watson Statistic Potmanteau Test
Normally Distributed	Goodness of Fit Test
All Properties	Joint Probabilities Log of Conditional Densities Cumulative-Sum Process Control

unlikely observations through residual monitoring.

Unlikely, and hence potentially transient, observations may be determined in a variety of ways. Since the residuals in the steady-state model are white and distributed normally with a mean of zero and the filter-computed variance, any violation of these assumptions would indicate that transient observations are being processed through the steady-state Kalman filter. Therefore, any of the statistical tests which are described previously as a method to accept the estimated model, can also be used to identify transient data. Those tests include the test of the residual mean, the tests of the residual whiteness, and the goodness-of-fit tests. In addition, some different approaches are apparent. Table 7 summarizes the truncation point selection criteria considered. The approaches that are not presented in the previous section are discussed. The cumulative-sum (cusum) process control algorithm is selected and tested in this research. The last part of this section discusses this application of the cusum algorithm. The section concludes with an example of the cusum algorithm.

Maybeck [57:230] depicts a method of residual monitoring technique based on the sum of natural logarithms of conditional densities of N_L residuals:

$$L_{N_L}(t_i) = c(t_i) - \frac{1}{2(\mathbf{HP} - \mathbf{H}^T + R)} \sum_{j=i-N_L+1}^i r^2(t_j)$$

The term $c(t_i)$ is a negative term independent of the observed residual values. This technique requires determining the group size N_L and a threshold value to determine significant deviations.

A second alternative is to apply a statistical hypothesis test on the variance. Hines and Montgomery [36:293] describe a test of hypothesis on the variance of a normal distribution. By using the theoretical mean of zero in calculating the sample variance of the residuals, this hypothesis test also detects a significant deviation from zero in the residual mean. The one-sided alternative is described because if the transient data does not fit the steady-state model, the resulting residual variance should be too large.

Therefore, for the hypothesis and alternative hypothesis are

$$H_0 : \sigma^2 = \sigma_0^2$$

$$H_A : \sigma^2 > \sigma_0^2$$

Set σ_0^2 equal to the filter-computed residual variance $\mathbf{HP}^{-1}\mathbf{H}^T + R$ of the steady-state filter, which is estimated using the least squares technique. The sample variance of the test residuals is approximated with the mean squared residuals:

$$S^2 = \frac{\sum_{n=1}^{N_{x^2}} r_n^2}{N_{x^2} - 1}$$

N_{x^2} is the number of observations processed for the test. The chi-square distributed test statistic with $N_{x^2} - 1$ degrees of freedom is

$$\chi_0^2 = \frac{(N_{x^2} - 1)S^2}{\sigma_0^2}$$

and the rule is to reject H_0 if $\chi_0^2 > \chi_{\alpha, N_{x^2}-1}^2$.

While a type I or α error is the probability of incorrectly rejecting the null hypothesis, a type II or β error is the probability of incorrectly accepting the hypothesis when it is

incorrect. Hines and Montgomery's text [36:606] contains operating characteristic curves which indicate that, with $\alpha = .05$, a sample size of $N_{\chi^2} = 30$ is necessary to have $\beta = .1$ when the true standard deviation is 1.5 times the hypothesized standard deviation. Similarly, with $\alpha = .01$ a sample size of $N_{\chi^2} = 40$ is needed. Therefore, besides determining critical values of the chi-square statistic, this hypothesis test provides insight into the desired sample size N_{χ^2} . This technique is very similar to Maybeck's since both use the sum of the residuals squared divided by the filter-computed variance. However, the advantage of this technique is that, in choosing the probabilities of type I and II errors, the sample size and critical value of the test statistic can be determined.

One additional point about hypothesis tests is pertinent. Even when the data is generated from the hypothesized linear system, the residuals fail the hypothesis test approximately 100α percent of the trials. However, during the transient phase, the data's relationship in time should be different than as predicted by the steady-state model. Thus, the transient values should produce statistical outliers in the residuals. Therefore, as with repeated application of any statistical test, the transient can be identified by a sequence of test results with a much higher failure rate than the specified α .

While hypothesis tests have the problem of increasing probability of type I error for repeated applications, process control algorithms can be designed to set the statistical frequency of false alarms [70:288]. After estimating the Kalman filter parameters for steady-state simulation output, the filter's residuals may be tracked to detect the onset of transient data. Of the many possible process control algorithms for monitoring sequences from which to choose, the cumulative-sum (cusum) quality control algorithm [70:291-295] is designed specifically to detect small changes in a sequence's mean. Since the objective is to identify transient simulation observations that bias the mean estimate, cusum seems to be an appropriate test. Furthermore, the assumptions of the cusum algorithm are that the sequence is white and normally distributed, with mean zero and known variance. The Kalman filter residual sequence follows these assumptions.

The remainder of this section describes the application of cusum to determine appropriate truncation points for discrete-event simulation output. A sensitivity analysis is presented to determine the number of residuals to aggregate in each cusum observation. After determining that each residual should constitute a separate cusum observation, the parameters for the cusum algorithm are selected. The section concludes with an example of using the cusum algorithm to select a truncation point.

The cusum approach is based on the concept that the cumulative sum of residuals should fluctuate near zero. The parameter Δ is the magnitude of the shift in the mean desired to be detected, H_{cs} is the cusum test's critical value, and N_{cs} is the number of residuals to average for an observation of the cusum test. For convenience, the residuals are standardized with $\tilde{r}_n = [\mathbf{HP}^{-1}\mathbf{H}^T + \mathbf{R}]^{-\frac{1}{2}}\mathbf{r}_n$. Thus, the average for the cusum test value is

$$\bar{\tilde{r}}_i = \frac{1}{N_{cs}} \sum_{n=N_{cs}(i-1)+1}^{N_{cs}i} \tilde{r}_n \text{ for } i = 1, 2, \dots, \frac{N}{N_{cs}}$$

The cusum test values are calculated with

$$S_H(i) = \max[0, \bar{\tilde{r}}_i - \frac{\Delta}{2} + S_H(i-1)]$$

$$S_L(i) = \max[0, -\bar{\tilde{r}}_i - \frac{\Delta}{2} + S_L(i-1)]$$

where $S_H(0) = S_L(0) = 0$. Considering $\frac{\Delta}{2}$ as a small tolerance, S_H represents the cumulative magnitude of a trend higher than the tolerance in the cusum observations. Likewise, $S_L(i)$ corresponds to the cumulative magnitude of a trend lower than the tolerance. If $S_H(i)$ or $S_L(i)$ exceeds H_{cs} , the sequence is "out-of-control" and is hypothesized to have encountered transient data. In addition, let N_H and N_L be the number of consecutive observations for which $S_H(i)$ and $S_L(i)$, respectively, have been positive. Then, a reasonable truncation point is N_H or N_L cusum observations prior to the failed test value.

Average Run Length (ARL) is the expected number of test observations processed before the cusum test indicates that the sequence is "out-of-control". If the process is in

Table 8. Cusum Average Run Length (ARL) for $\Delta = 1$

Shift in Mean (Multiples of $\sigma_{\bar{x}}$)	$H_{cs} = 4$	$H_{cs} = 5$
0	168	465
0.25	74.2	139
0.50	26.6	38.0
0.75	13.3	17.0
1.00	8.38	10.4
1.50	4.75	5.75
2.00	3.34	4.01
2.50	2.62	3.11
3.00	2.19	2.57
4.00	1.71	2.01

Note: Table reprinted from Montgomery [70:295]

steady state, the ARL would indicate the expected number of tests performed between occurrence of false alarms. False alarms occur when the algorithm indicates that transient data has been encountered when actually only steady-state data has been processed. In general, a large ARL is desired when the steady-state data is being encountered, and a short ARL when transient data is processed. By selection of Δ , H_{cs} , and N_{cs} , the ARL can be varied.

The following discussion examines the sensitivity of varying the number of residuals in a cusum observation. The fastest detection of a small shift in the mean is with only one residual in the cusum observation, $N_{cs} = 1$. Montgomery [70:295] presents Table 8 of ARL values when $H_{cs} = 4$ or $H_{cs} = 5$ with $\Delta = 1$. The table assumes the cusum observations are standard normal random variables. Therefore, Montgomery selects these values with the objective of detecting a one standard deviation shift in the mean of the process.

Lucas [55] describes a methodology for selecting the parameters for a cusum control scheme. He recommends that the "in-control" expected ARL be about 500 cusum observations. In addition, the cusum test for this application is designed to detect a shift in mean of the standardized residuals of one half standard deviation, so $\Delta = \frac{1}{2}\sigma_{\bar{x}} = \frac{1}{2}$.

Table 9. Cusum Batch Size N_{cs} Effect on Average Run Length (ARL)

Parameters			Shift in the Residual Mean (Multiples of $\sigma_{\bar{r}}$)						
N_{cs}	Δ	H_{cs}	0	0.25	0.50	0.75	1.00	1.50	2.00
1	0.5	8.59	499/499	94/94	30.9/30.9	17.5/17.5	12.1/12.1	7.6/7.6	5.5/5.5
4	1.0	5.07	499/1996	38.8/155.2	10.5/42.0	5.8/23.2	4.1/16.4	2.6/10.4	2.0/8.0
9	1.5	3.55	497/4473	22.1/198.9	5.5/49.5		2.2/19.8		
16	2.0	2.67	504/8064	14.6/233.6	3.4/54.4	1.9/30.4	1.4/22.4		
36	3.0	1.70	492/17,712	8.1/291.6	1.8/64.8		$\approx 1/36$		

Note: Values are expected number of cusum observations/expected number of residuals

If the variance of the observations used for the cusum test is reduced, the ARL increases. The variance can be reduced significantly by using the average of a batch of residuals for each cusum observation rather than individual residuals. While fewer cusum observations are necessary to detect a change in the mean, each of these observations contains more residuals. Lucas' graphs [55] show that H_{cs} varies approximately linearly with $\log(\text{ARL})$. Using this interpolation scheme and his table of values, the tradeoff between the cusum batch size N_{cs} and the ARL can be seen in Table 9. Lucas provides values only for the sample sizes shown.

As the cusum batch size N_{cs} increases from 1 to 4, the standard deviation of each cusum observation is decreased by $\frac{1}{\sqrt{N}} = \frac{1}{2}$. Thus the magnitude of $\sigma_{\bar{r}}$ with $N_{cs} = 1$ equals the magnitude of $2\sigma_{\bar{r}}$ with $N_{cs} = 4$, as seen by comparing Table 8 and Table 9. From Table 9, the expected ARL of $N_{cs} = 4$ and a mean shift of $0.25\sigma_{\bar{r}}$ is 38.8 cusum observations. Furthermore, $0.25\sigma_{\bar{r}} = 0.5\sigma_{\bar{r}}$ when $N_{cs} = 4$. In Table 8, expected ARL for a shift in the mean of $0.5\sigma_{\bar{r}}$ is 38.0 with $H_{cs} = 5$. The difference between the two values can be explained by the slight change in the critical value H_{cs} from 5.07 to 5.0.

Increasing the batch size for a cusum observation reduces the variance of the cusum observation. Therefore, large batches can detect a shift in the residual mean in fewer cusum observations. However, the advantage of increased batch size appears to be outweighed by the additional residuals required in each cusum observation. In every case in Table 9, a batch size N_{cs} larger than one requires more residuals to detect the same magnitude of shift

in the mean. Therefore, for this application, a batch size of one, $N_{cs} = 1$, is used.

With $N_{cs} = 1$, selection of the magnitude of the shift to detect Δ and the desired "in-control" ARL determines the cusum critical value H_{cs} . Lucas [55] recommends that the "in-control" ARL be about 500 cusum observations, and Gan [26] presents graphs from which to determine the H_{cs} given Δ and the ARL. The applied cusum test is designed to detect a shift in mean of the standardized residuals of one half a standard deviation, so $\Delta = \frac{1}{2}\sigma_{\bar{r}} = \frac{1}{2}$. The resulting critical value H_{cs} is 8.6. The "in-control" ARL is 500, and the theoretical "out-of-control" ARL for a shift of Δ in the residuals is about 31 observations. In actual application, results may be degraded from these theoretical values due to modeling assumptions and estimation errors.

After the cusum test indicates an "out-of-control" signal, a simple test for a false alarm is to reset S_H and S_L to zero without resetting N_H or N_L . If the data has really changed from the steady-state model, the cusum algorithm should quickly indicate "out-of-control" again. Therefore, if a second "out-of-control" signal is calculated within 15 observations, the truncation point is selected. The observation counters N_H and N_L are not reset so the selected truncation point \hat{n}_o remains where the residuals sequence first indicate a systematic change.

Let the parameter estimation have completed on the $(N - J)$ th observation, continue processing backwards through the data $\{y_{N-J-1}, y_{N-J-2}, \dots, y_1\}$ with the estimated Kalman filter. For convenience, the residuals are transformed to standard normal random variables, $\bar{r}_{N-j} = [\mathbf{H}\mathbf{P}^{-}\mathbf{H}^T + \mathbf{R}]^{-\frac{1}{2}}r_{N-j}$. The applied cusum test values are

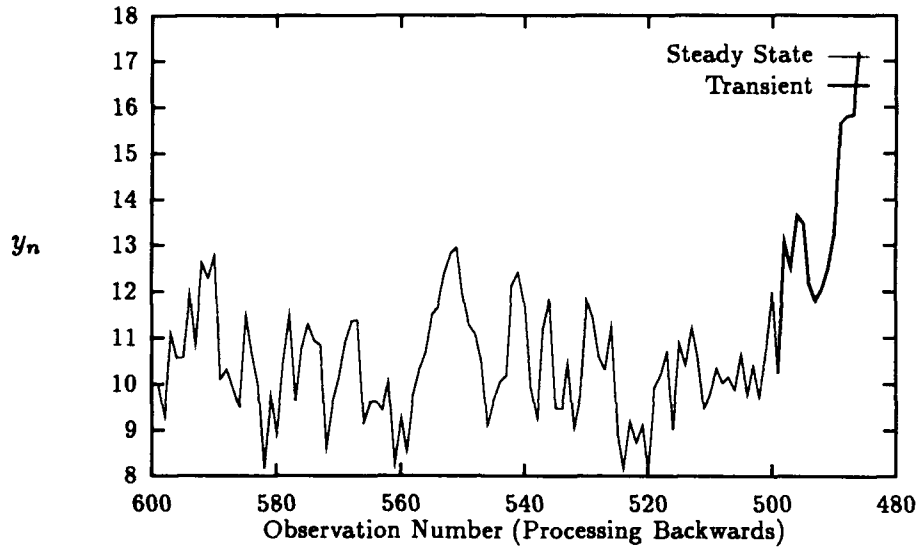
$$S_H(j) = \max[0, \bar{r}_{N-j} - \frac{\Delta}{2} + S_H(j-1)] \quad (59)$$

$$S_L(j) = \max[0, -\bar{r}_{N-j} - \frac{\Delta}{2} + S_L(j-1)] \quad (60)$$

where $S_H(J) = S_L(J) = 0$ and $j = J+1, J+2, \dots, N$

If the cusum test for a high trend $S_H(j)$ is above the critical value H_{cs} , then the cusum test

Figure 10. Simulation Output

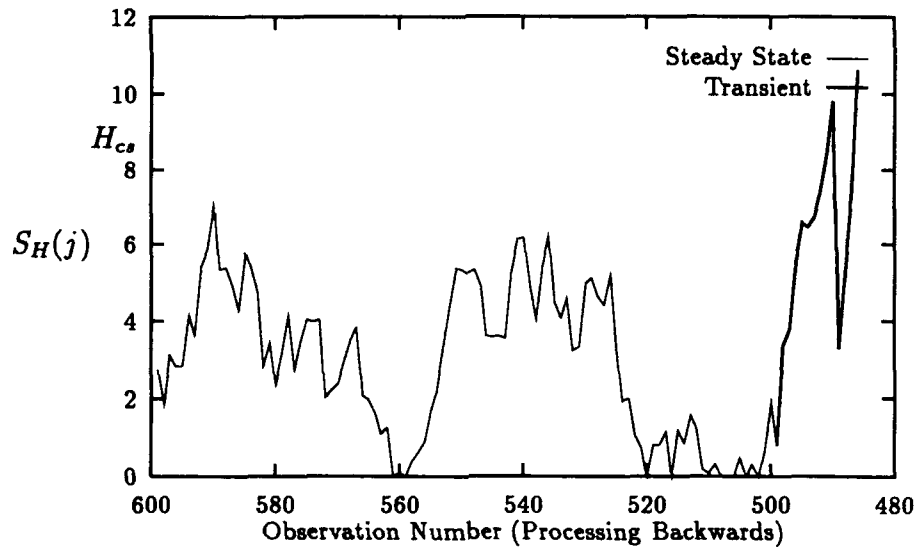


for a low trend $S_L(j)$ must be zero. If $S_H(j) > H_{cs}$, then $S_L(j) = 0$ and $N_L = 0$. The reverse, if $S_L(j) > H_{cs}$, then $S_H(j) = 0$ and $N_H = 0$, is also true. Therefore, one possible estimated truncation point \hat{n}_0 is where the cusum observations began a trend that leads to exceeding the critical value. This truncation point can be calculated for both the high and low "out-of-control" signals with

$$\hat{n}_0 = N - j + N_H + N_L \quad \text{for } S_H(j) > H_{cs} \text{ or } S_L(j) > H_{cs} \quad (61)$$

Figure 10 shows sample simulation output from an AR(1) process. The mean of the process has a step shift from 10 to 12 at observation 500. Figure 11 shows the corresponding cusum test statistic S_H . The test statistic is reset when it exceeded the critical value of 8.6. Upon the second crossing within 15 cusum tests, the truncation point is estimated at N_H observations prior where $S_H(j)$ first began the sequence of consecutive positive values. As shown in Figure 11, the selected truncation point is observation 502 when the actual transient ended at the 500th observations. The simulation output prior to the truncation point is not

Figure 11. Cusum Test Statistic $S_H(j)$



shown.

This section has developed a truncation point selection algorithm based on residual monitoring. For the estimated Kalman filter, the simulation output sequence is processed through the Kalman filter backwards. The residuals are monitored by the cusum statistical process control algorithm. When the cusum algorithm indicates that sequence is “out-of-control”, the cusum statistics are reset to test for a false alarm. Upon a second “out-of-control” signal within a small number of observations, the algorithm assumes transient data is being processed. The truncation point is selected where the cusum statistics started the trend leading to the first “out-of-control” signal. The next section describes the specific steps in the algorithm.

4.5 Residual Monitoring Truncation-Point Selection Algorithm Steps

Many alternative methods have been discussed for developing a truncation-point selection algorithm. This section summarizes the techniques that are tested and details the specifics of their implementation.

This technique is a post-processing approach, meaning that the simulation model is run and the univariate output is stored in a file before beginning the truncation-point algorithm. An AR(1) or AR(2) system model with measurement noise is fit to the simulation output with least squares estimation. Applying the Kalman filter, the residuals are monitored with the cusum quality control algorithm to identify an appropriate truncation point \hat{n}_0 . The steps of the truncation-point selection algorithm are:

Step 1. Using the last 250 of the simulation observations, $\{y_N, y_{N-1}, \dots, y_{N-249}\}$, make an initial estimate of $\hat{\mu}_y$:

$$\hat{\mu}_y = \frac{1}{250} \sum_{j=0}^{249} y_{N-j}$$

Step 2. Set up a bank of Kalman filters with each filter having a different combination of the discrete parameters, $\hat{\mu}_y, \hat{\phi}_1, \hat{\phi}_2$, and \hat{k} . Discretize the parameters in increments of 0.1. The mean estimates vary from -0.3 to +0.3 of $\hat{\mu}_y$ estimated in step 1. \hat{k} varies from 0.0 to 1.0. The autoregressive coefficients $\hat{\phi}_1$ and $\hat{\phi}_2$ are varied across the stable range in increments of 0.1. Exclude $\hat{\phi}_1 = 0$ since this is the same as $\hat{k} = 0$. For AR(2) formulation, only stable combinations of $\hat{\phi}_1$ and $\hat{\phi}_2$ [11] are used, resulting in a bank of 3,773 or fewer filters. Also for each AR(2) filter, calculate an associated \mathbf{K} using Equation (51). The state estimates $\hat{\mathbf{x}}(t_0^+)$ are initialized with zeros.

Step 3. For each Kalman filter in the bank, beginning with the last observation y_N , process the observations. Propagate using Equations (22) and update with Equations (47), (26) and (27). For each filter, maintain a cumulative sum of the squared residuals. For each observation after the $(N - 250)$ th, if the filter with the minimum sum of squared residuals also meets the criterion in Equation (58), select the parameters of that filter as the least square estimates. If no parameters are estimated prior to encountering the end of the simulation data, steady state is assumed not to have been achieved in that output sequence.

Optional Step. To reduce computer run time, particularly for Monte Carlo tests, after observations y_{N-100} , y_{N-200} , and y_{N-300} are processed, turn off any Kalman filters

with a sum of squared residuals greater than 1.5 times the minimum sum of squared residuals.

Step 4. Assume the model estimation in Step 2 completes on the $(N - J)$ th observation.

The cusum algorithm, Equations (59) and (60), is initiated using the residuals of the least squares Kalman filter. Set $S_H(J) = S_L(J) = 0$ and continue processing backwards beginning with the first observation not used in least squares estimation. When either of the cusum test values, $S_H(j)$ or $S_L(j)$, exceed the control limit, $H_{cs} = 8.6$, the point where the cusum test value first became positive, as calculated with Equation (61), is considered a potential truncation point n_0 . This point is tested for a false alarm in the control algorithm by resetting the cusum test values, $S_H(j)$ and $S_L(j)$, to zero. If $S_H(j)$ or $S_L(j)$ exceeds H_{cs} again within 15 observations, \hat{n}_o is selected as the truncation point. Otherwise, the cusum algorithm is continued until a new potential truncation point is determined. If no truncation point is selected in the entire stream, the algorithm failed.

Optional Step. Determine \hat{Q}_d , \hat{R} , $\hat{\sigma}_v$, or $\hat{\sigma}_{\hat{\mu}_v}$ with the least squares parameter estimates and the approximate residual sample variance.

4.6 Monte Carlo Results

A Monte Carlo analysis is presented to show the effectiveness of using the Kalman filter residual monitoring approach to determine the appropriate data truncation point. This analysis consists of a loop that generates a set of data, applies the truncation-point selection algorithm, evaluates the selected point, and collects the evaluation statistics.

Although there is not a widely accepted or proven analytical approach for simulation initial data truncation, Welch's [101] graphical approach is the most commonly applied technique. The disadvantages of Welch's technique is that it requires a subjective assessment, and therefore, it is difficult to test in a Monte Carlo analysis.

If the generated data sets have a known transition point from the transient phase to the steady-state phase, two performance measures are the number of observations between the known and the selected truncation point and the associated MSE. Schruben's [87] univariate test for presence of initial bias, described in Section 2.4.1, can also be used to determine if the truncation point deleted enough data.

Wilson and Pritsker [104] state that the true test for any technique is the effect on our ability to draw inferences from the data. They recommend forming confidence intervals on systems with analytical steady-state values. Then alternative techniques can be compared by their resulting confidence interval half-widths and their actual coverage rates.

The specific performance measures for the Monte Carlo analysis are as follows:

1. The minimum, maximum, bias, and MSE of the selected truncation points.
2. The coverage rate for confidence intervals calculated with nonoverlapping and overlapping batch means [69] technique using Fishman's algorithm [22, 23] to determine batch size.
3. The percentage of truncation sequences that pass Schruben's test for no initial bias [87].

Besides knowing the theoretical process mean of the generated data, the process variance is also known and is used to evaluate the variance estimate.

The data used to test the Kalman filters truncation-point selection algorithm is generated with time-series models. For these models, the data is generated so that the onset of steady state is known. The generated transient phases are caused by four types of changes in the simulation output mean. The four transient types are a jump discontinuity, a ramp change, a decaying exponential, or a decaying sinusoidal. The test data sets are generated from AR(2) and ARMA(1,1) model sequences with an induced transient. The Monte Carlo results are divided into two corresponding sections.

Table 10. AR(2) Parameters for Generated Data

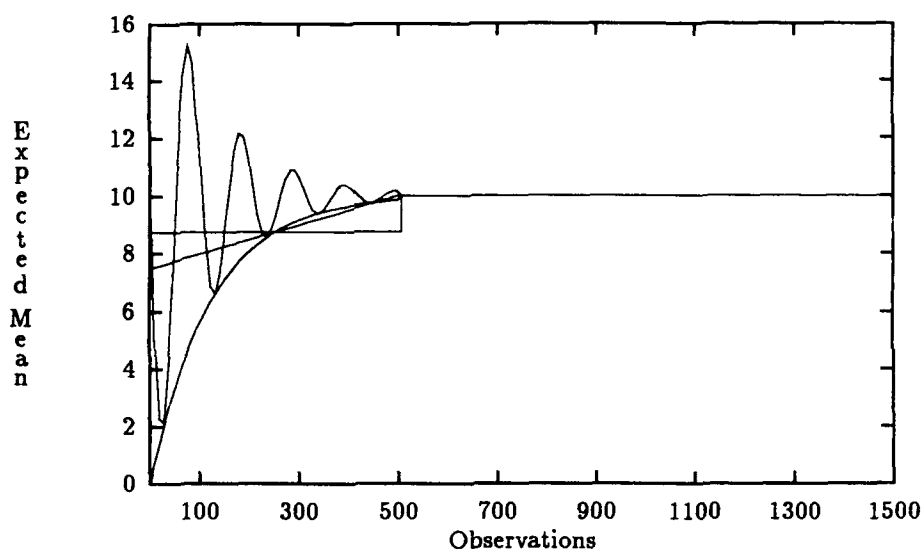
ϕ_1	ϕ_2	$\rho_y(1)$	σ_y^2	$\approx N\sigma_{\hat{\mu}_y}^2$
+1.5	-0.8	+0.83	9.09	11.11
+1.0	-0.5	+0.67	2.40	4.00
+0.5	+0.3	+0.71	2.24	25.00
+0.5	-0.3	+0.38	1.29	1.56
-0.5	+0.3	-0.71	2.24	0.69
-0.5	-0.3	-0.38	1.29	0.31
-1.0	-0.5	-0.67	2.40	0.16
-1.5	-0.8	-0.83	9.09	0.09

4.6.1 Autoregressive Test Cases. A Monte Carlo analysis is conducted, and both the AR(1) with measurement noise and AR(2) with measurement noise formulations are tested. For each combination of parameters and conditions, 1,000 pure AR(2) data sequences are generated. The transients are induced by varying the mean of the sequence. Four types of transients, step change, ramp, exponential decay, and sinusoidal decay, are used. Each of these transients are tested at four different magnitudes or rates of change.

Pure AR(2) data is generated with Equation (43). The specific parameters used to generate the AR(2) data, shown in Table 10, are selected to produce output with a wide range of autocorrelation and partial autocorrelation curves. For each of the generated data sets, the variance of the dynamic noise Q_d is 1.0, and no measurement noise is added, so $R = 0$. The theoretical autocorrelation coefficient at lag one, $\rho_y(1) = \frac{\phi_1}{1-\phi_2}$, indicates that most of the selected models have relatively high autocorrelation. The sequence variances σ_y^2 , given in Equation (54), have a wide range. The last column reports the sample size N times Fishman's approximate variance of the estimated mean $\sigma_{\hat{\mu}_y}^2$ [23:249] calculated with the true parameters in Equation (11).

Transients for the time series models are induced by changing the expected value of the generated sequence. Four types of transients, depicted in Figure 12, are used. The first transient type is a simple step change in the mean. The next type is a ramp or linear change in the mean. The final two transient types are an exponential decay and an exponentially

Figure 12. Types of Transients



decaying sinusoidal in the mean of the output. For each run, 500 transient observations and 1,000 steady-state values are generated. In order to compare between transient types, the magnitude of the change in the expected value at the middle of the transient, observation 250, is equal for each of the transient types. This shift in the magnitude of the expected value of the sequence, labeled $\Delta\mu(\sigma_y)$ in the tables, is expressed in multiples of the standard deviation of the generated data σ_y .

Statistics, shown in Table 11, are collected on the technique's model estimation and residual tracking. The results for both of the Kalman filter formulations are shown. The top of the table are the results for the Kalman filter with an AR(1) with measurement noise formulation, and the bottom half of the table reports the results for the Kalman filter with an AR(2) with measurement noise formulation. (These two models are discussed on page 81.) The column 'Fail' indicates the frequency with which the least squares technique failed to estimate a steady state model. Failure to estimate a model occurred when the filter with the minimum sum of squared residuals never met the test criterion of near-zero sum of residuals, Equation (58). When the AR(2) data is generated with ϕ_1 of -1.0 or -1.5, the estimation

strategy fails to meet this criterion for over a fourth of the runs.

The next set of columns in Table 11 are statistics on the number of simulation observations used to estimate the system parameters. The last 250 observations are assumed steady state and used to obtain an initial estimate of the mean, so the least squares estimation is not permitted to stop until after 250 observations have been processed. Therefore, the minimum occurs at just greater than 250. The maximum number of observations used is just prior to 1500 since that is the number of available observations. The average number of observations used to estimate the model for the last four cases is significantly higher than the first four cases. This may be because the last four cases have negative first autocorrelation coefficients.

The final set of columns reports statistics on the number of "out-of-control" signals generated by the cusum algorithm between model estimation and truncation point estimation. After the model is estimated, two cusum "out-of-control" signals within 15 observations are necessary to declare the transient.

The Kalman filter modeled as an AR(1) with measurement noise formulation and the Kalman filter modeled as an AR(2) with measurement noise model require approximately the same number of observations to estimate a model. In addition, no difference in the number of cusum alarms is apparent. Since the generated sequence consisted of 500 transient observations and 1000 steady-state observations, many of estimated parameters are based on some transient data. In the last four cases, which have negatively correlated sequences, the estimation algorithm generally requires more observations, fails to estimate a steady-state model at a higher frequency, and has fewer cusum alarms. The fewer cusum alarms are probably a result of fewer remaining data values after model estimation.

Statistics are also collected on the least squares estimated parameters, $\hat{\mu}_y$, $\hat{\phi}_1$, $\hat{\phi}_2$, and \hat{k} . Table 12 shows the Kalman filter parameter estimates for the AR(1) formulation. The mean estimate $\hat{\mu}_y$ is very near the steady-state mean of 10.0. Since the data is actually AR(2), the value of $\hat{\phi}_1$ attempts to account for the correlation produced by ϕ_1 and ϕ_2 .

Table 11. Kalman Filter Model Estimation Summary

Data Parameters		Fail	Observations Used To Estimate				Cusum Alarms			
			Min	Max	Avg	St Dev	Min	Max	Avg	St Dev
AR(2) Data with AR(1) with Measurement Noise Model										
$\phi_1 = 1.5$	$\phi_2 = -0.8$	0.007	252	1447	384.7	179.6	1	24	4.7	2.6
$\phi_1 = 1.0$	$\phi_2 = -0.5$	0.000	252	1347	303.3	70.4	2	21	4.8	2.5
$\phi_1 = 0.5$	$\phi_2 = 0.3$	0.045	252	1460	411.8	239.2	0	21	5.6	2.8
$\phi_1 = 0.5$	$\phi_2 = -0.3$	0.004	254	1497	508.7	301.9	0	22	4.5	2.4
$\phi_1 = -0.5$	$\phi_2 = 0.3$	0.099	265	1499	808.0	384.5	0	17	4.0	2.3
$\phi_1 = -0.5$	$\phi_2 = -0.3$	0.174	272	1499	1057.4	351.2	0	18	3.2	2.0
$\phi_1 = -1.0$	$\phi_2 = -0.5$	0.273	296	1499	1191.0	244.2	0	17	2.8	1.8
$\phi_1 = -1.5$	$\phi_2 = -0.8$	0.316	305	1499	1197.8	174.1	0	18	3.0	2.1
AR(2) Data with AR(2) with Measurement Noise Model										
$\phi_1 = 1.5$	$\phi_2 = -0.8$	0.001	252	1432	300.2	57.9	2	18	6.3	2.5
$\phi_1 = 1.0$	$\phi_2 = -0.5$	0.003	254	1429	366.2	147.5	2	19	5.6	2.4
$\phi_1 = 0.5$	$\phi_2 = 0.3$	0.025	252	1496	363.8	183.5	0	22	5.8	2.7
$\phi_1 = 0.5$	$\phi_2 = -0.3$	0.017	256	1499	555.9	322.6	0	21	4.9	2.5
$\phi_1 = -0.5$	$\phi_2 = 0.3$	0.066	261	1498	803.4	382.5	0	19	3.7	2.1
$\phi_1 = -0.5$	$\phi_2 = -0.3$	0.210	273	1499	1055.1	349.1	0	19	3.1	1.9
$\phi_1 = -1.0$	$\phi_2 = -0.5$	0.328	299	1499	1190.4	243.3	0	16	2.5	1.4
$\phi_1 = -1.5$	$\phi_2 = -0.8$	0.461	305	1499	1189.2	174.6	0	8	2.1	0.5

Note: Each row represents 16,000 runs: 1,000 for each transient type and magnitude.

Table 12. AR(1) with Measurement Noise Model Parameter Estimation Summary

Data Parameters		$\hat{\mu}_y$		$\hat{\phi}_1$		\hat{k}	
		$\hat{\mu}_{\hat{\mu}_y}$	$\hat{\sigma}_{\hat{\mu}_y}$	$\hat{\mu}_{\hat{\phi}_1}$	$\hat{\sigma}_{\hat{\phi}_1}$	$\hat{\mu}_{\hat{k}}$	$\hat{\sigma}_{\hat{k}}$
AR(2) Data with AR(1) with Measurement Noise Model							
$\phi_1 = 1.5$	$\phi_2 = -0.8$	10.00	0.19	0.80	0.02	1.00	0.00
$\phi_1 = 1.0$	$\phi_2 = -0.5$	10.00	0.12	0.67	0.04	1.00	0.00
$\phi_1 = 0.5$	$\phi_2 = 0.3$	10.00	0.28	0.86	0.05	0.61	0.08
$\phi_1 = 0.5$	$\phi_2 = -0.3$	9.99	0.08	0.40	0.05	1.00	0.00
$\phi_1 = -0.5$	$\phi_2 = 0.3$	9.96	0.10	-0.89	0.03	0.55	0.11
$\phi_1 = -0.5$	$\phi_2 = -0.3$	9.95	0.08	-0.19	0.44	0.89	0.29
$\phi_1 = -1.0$	$\phi_2 = -0.5$	9.92	0.08	-0.62	0.07	0.99	0.05
$\phi_1 = -1.5$	$\phi_2 = -0.8$	9.89	0.12	-0.79	0.03	1.00	0.00

Note: Each row represents the runs that did not fail to estimate a model from 16,000 runs.

Table 13. AR(1) with Measurement Noise Model Variances Estimation Summary

Data Parameters		\hat{Q}_d		\hat{R}		$\hat{\sigma}_y$		σ_y
		$\hat{\mu}_{\hat{Q}_d}$	$\hat{\sigma}_{\hat{Q}_d}$	$\hat{\mu}_{\hat{R}}$	$\hat{\sigma}_{\hat{R}}$	$\hat{\mu}_{\hat{\sigma}_y}$	$\hat{\sigma}_{\hat{\sigma}_y}$	
AR(2) Data with AR(1) with Measurement Noise Model								
$\phi_1 = 1.5$	$\phi_2 = -0.8$	2.77	0.46	0.00	0.00	2.80	0.45	3.02
$\phi_1 = 1.0$	$\phi_2 = -0.5$	1.33	0.14	0.00	0.00	1.57	0.11	1.55
$\phi_1 = 0.5$	$\phi_2 = 0.3$	0.44	0.11	0.39	0.08	1.51	0.41	1.50
$\phi_1 = 0.5$	$\phi_2 = -0.3$	1.11	0.09	0.00	0.01	1.15	0.07	1.14
$\phi_1 = -0.5$	$\phi_2 = 0.3$	0.39	0.09	0.55	0.32	1.56	0.13	1.50
$\phi_1 = -0.5$	$\phi_2 = -0.3$	1.02	0.38	0.16	0.43	1.19	0.11	1.14
$\phi_1 = -1.0$	$\phi_2 = -0.5$	1.51	0.20	0.03	0.11	1.59	0.08	1.55
$\phi_1 = -1.5$	$\phi_2 = -0.8$	3.20	0.55	0.00	0.01	2.93	0.20	3.02

Note: Each row represents the runs that did not fail to estimate a model from 16,000 runs.

From these estimates for $\hat{\mu}_y$, $\hat{\phi}_1$, $\hat{\phi}_2$, and \hat{k} , the noise variances, \hat{Q}_d and \hat{R} , are determined, as shown in Table 13. Finally, $\hat{\sigma}_y$ is calculated by taking the square root of the variance calculated with the estimated parameters in Equation (55). The theoretical standard deviation σ_y for the generated AR(2) data is presented for comparison. Where necessary, the estimates adjusted for the "lack of fit" with inflated estimates of the dynamic noise variance \hat{Q}_d (greater than $Q_d = 1.0$) and by adding measurement noise variance \hat{R} . Since $\hat{\sigma}_y$ is based on the bias estimates of Q_d and R , these estimates also are biased.

Table 14. AR(2) with Measurement Noise Model Parameter Estimation Summary

Data Parameters	$\hat{\mu}_y$		$\hat{\phi}_1$		$\hat{\phi}_2$		\hat{k}	
	$\hat{\mu}_{\hat{\mu}_y}$	$\hat{\sigma}_{\hat{\mu}_y}$	$\hat{\mu}_{\hat{\phi}_1}$	$\hat{\sigma}_{\hat{\phi}_1}$	$\hat{\mu}_{\hat{\phi}_2}$	$\hat{\sigma}_{\hat{\phi}_2}$	$\hat{\mu}_{\hat{k}}$	$\hat{\sigma}_{\hat{k}}$
AR(2) Data with AR(2) with Measurement Noise Model								
$\phi_1 = 1.5$ $\phi_2 = -0.8$	10.00	0.20	1.54	0.05	-0.84	0.04	0.90	0.00
$\phi_1 = 1.0$ $\phi_2 = -0.5$	10.00	0.11	1.08	0.06	-0.58	0.06	0.90	0.00
$\phi_1 = 0.5$ $\phi_2 = 0.3$	10.00	0.28	0.53	0.10	0.26	0.08	0.87	0.08
$\phi_1 = 0.5$ $\phi_2 = -0.3$	9.99	0.08	0.58	0.07	-0.35	0.07	0.88	0.04
$\phi_1 = -0.5$ $\phi_2 = 0.3$	9.97	0.07	-0.47	0.13	0.35	0.13	0.88	0.05
$\phi_1 = -0.5$ $\phi_2 = -0.3$	9.95	0.08	-0.62	0.10	-0.39	0.09	0.72	0.22
$\phi_1 = -1.0$ $\phi_2 = -0.5$	9.93	0.08	-1.09	0.06	-0.57	0.05	0.81	0.12
$\phi_1 = -1.5$ $\phi_2 = -0.8$	9.92	0.08	-1.53	0.04	-0.83	0.04	0.85	0.06

Note: Each row represents the runs that did not fail to estimate a model from 16,000 runs.

Table 15. AR(2) with Measurement Noise Model Variance Estimation Summary

Data Parameters	\hat{Q}_d		\hat{R}		$\hat{\sigma}_y$		σ_y
	$\hat{\mu}_{\hat{Q}_d}$	$\hat{\sigma}_{\hat{Q}_d}$	$\hat{\mu}_{\hat{R}}$	$\hat{\sigma}_{\hat{R}}$	$\hat{\mu}_{\hat{\sigma}_y}$	$\hat{\sigma}_{\hat{\sigma}_y}$	
AR(2) Data with AR(2) with Measurement Noise Model							
$\phi_1 = 1.5$ $\phi_2 = -0.8$	0.70	0.06	0.10	0.01	2.91	0.41	3.02
$\phi_1 = 1.0$ $\phi_2 = -0.5$	0.78	0.06	0.10	0.01	1.54	0.10	1.55
$\phi_1 = 0.5$ $\phi_2 = 0.3$	0.82	0.13	0.13	0.08	1.44	0.23	1.50
$\phi_1 = 0.5$ $\phi_2 = -0.3$	0.85	0.10	0.12	0.04	1.15	0.07	1.14
$\phi_1 = -0.5$ $\phi_2 = 0.3$	0.89	0.14	0.12	0.05	1.51	0.13	1.50
$\phi_1 = -0.5$ $\phi_2 = -0.3$	0.69	0.23	0.36	0.37	1.17	0.06	1.14
$\phi_1 = -1.0$ $\phi_2 = -0.5$	0.76	0.13	0.28	0.25	1.57	0.07	1.55
$\phi_1 = -1.5$ $\phi_2 = -0.8$	0.83	0.11	0.26	0.18	3.05	0.30	3.02

Note: Each row represents the runs that did not fail to estimate a model from 16,000 runs.

Table 14 shows that when the data is from an AR(2) process, the Kalman filter estimates $\hat{\mu}_y$, $\hat{\phi}_1$ and $\hat{\phi}_2$ are very accurate. The associated variance estimates are presented in Table 15. The estimate for the variance of the dynamic driving noise \hat{Q}_d underestimates the true value $Q_d = 1.0$, but the measurement noise variance \hat{R} is greater than zero to compensate.

Generally, the model estimation is completed significantly prior to the transient phase.

The parameter estimation results are similar across the four transient types. Therefore, the model and parameter estimation tables are summarized across the four types of transients. However, the results on estimating the end of the transient phase are very dependent on the type of transient and are tabulated separately for each transient.

The performance measures are selected to evaluate the effectiveness of the Kalman filter residual monitoring approach to select an appropriate truncation point \hat{n}_o . Since the transient phase ends in the generated sequence at the 500th observation, the accuracy is evaluated in terms of this true truncation point n_o . Tables 16, 17, 18, and 19 report statistics that compare the estimated truncation point \hat{n}_o with the true truncation point n_o . Each transient type and mean shift magnitude combination is tested with 1,000 Monte Carlo runs.

The column 'Fail' indicates the frequency with which no truncation point is selected after the model is estimated. This type of failure generally occurs only in a small percentage of the runs, except for the case where the transient is a step change with the smallest magnitude of shift, $\frac{1}{2}\Delta\mu_y(\sigma_y)$. This high failure rate to select a truncation point occurs because of a combination of two transient characteristics. First, the step transient is the only one of the four transient types for which the magnitude of shift did not continue to increase as the algorithm approaches the beginning of the simulated output. Second, the amount of shift in the mean, $\frac{1}{2}\sigma_y$, is too small for the cusum algorithm to detect. The cusum algorithm is designed to detect a shift of one half a standard deviation in the magnitude of the residuals $\frac{1}{2}\sigma_{r_n}^2$. The transient with the smallest change is apparently too small to detect in many output sequences.

Negative truncation-point errors ($\hat{n}_o - n_o < 0$) indicate that transient data is retained in the truncated sequence, and positive errors ($\hat{n}_o - n_o > 0$) result from excessive truncation. In other words, negative truncation-point errors indicate the average number of transient observations that remain in the truncated sequence. In contrast, positive truncation errors show the number of steady-state observations that are truncated. For example, as shown under the column label "Avg" in Table 16, the average truncation-point error for the transient type "2.0 Ramp" is -29.3. For this case, the average selected truncation-point is 470.7

while the generated end of transient is at 500. For the sixteen data types and transient combinations tested, the AR(1) formulation results in average selected truncation-points n_0 significantly in the transient.

Tables 20, 21, 22, and 23 report similar truncation error statistics for the AR(2) with measurement noise formulation applied to the same AR(2) time series models and transient types. Generally, the AR(2) formulation improves the results when compared to the AR(1) results. It is not surprising that the AR(2) formulation is able to better fit and detect a change in AR(2) data. The improvement is apparent in smaller average truncation errors. The standard deviation of the truncation points and the Mean Square Error (MSE) sometimes increases and occasionally decreases when comparing the AR(1) to the AR(2) formulation.

In general, the truncation-point selection algorithm picks points near the end of the induced transient. As the the shift in the transient mean decreases from $2\sigma_y$ to only $\frac{1}{2}\sigma_y$, the selected truncation point \hat{n}_0 includes more transient data, as anticipated. The exponentially decaying sinusoidal is the most difficult transient to identify, probably since the transient varies about the steady-state mean.

Since the general objective of steady state analysis is to draw inferences about the output distribution, appropriate performance measures should reflect the effect of the estimated truncation point on our ability to construct confidence intervals [44, 52, 104, 105]. Since the sampling distribution is typically assumed to be symmetric about the point estimator, the half width of the confidence interval, along with realized coverage, is generally used for comparisons. The effectiveness of confidence interval construction in terms of coverage and half widths is reported for the nonoverlapping batch means (NOBM) and the overlapping batch means (OBM) [69] techniques. (See page 26 for description.) Since the truncated sequence should be without initial bias, the final criterion selected is the ability to pass Schruben's test for no initial bias [87]. For comparison, the same performance measures are reported for the sequence truncated at the induced true truncation point n_0 . A nominal rate of 0.9 is arbitrarily selected for each criterion.

Table 16. AR(1) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = 1.5$ and $\phi_2 = -0.8$							
2.0	Step	0.000	-15	724	21.2	88.1	8211.9
2.0	Ramp	0.000	-165	708	-29.3	100.4	10934.9
2.0	Exp	0.000	-53	717	21.1	90.5	8640.2
2.0	Sine	0.004	-427	710	15.8	101.9	10640.7
1.5	Step	0.000	-43	727	27.6	105.5	11901.5
1.5	Ramp	0.000	-248	733	-47.0	112.6	14883.2
1.5	Exp	0.000	-231	705	16.4	95.4	9378.7
1.5	Sine	0.004	-437	709	-3.1	97.5	9516.1
1.0	Step	0.003	-443	690	22.3	101.3	10761.2
1.0	Ramp	0.000	-318	732	-85.7	128.2	23784.7
1.0	Exp	0.000	-306	715	-61.4	117.9	17676.0
1.0	Sine	0.002	-382	717	-119.7	138.1	33399.1
0.5	Step	0.072	-470	739	-52.7	158.4	27873.8
0.5	Ramp	0.006	-462	700	-190.2	158.1	61164.2
0.5	Exp	0.000	-392	723	-193.5	148.9	59598.7
0.5	Sine	0.000	-382	699	-237.9	160.1	82236.0
AR(2) Data with $\phi_1 = 1.0$ and $\phi_2 = -0.5$							
2.0	Step	0.000	-2	40	2.8	5.5	37.6
2.0	Ramp	0.000	-159	22	-78.9	32.1	7256.4
2.0	Exp	0.000	-148	81	-39.3	35.2	2783.0
2.0	Sine	0.000	-201	469	-88.2	50.5	10325.0
1.5	Step	0.000	-12	45	2.8	5.6	39.0
1.5	Ramp	0.000	-220	2	-113.4	38.4	14325.8
1.5	Exp	0.000	-211	37	-113.6	46.6	15082.4
1.5	Sine	0.000	-260	-5	-169.0	43.5	30457.8
1.0	Step	0.000	-222	72	-7.6	24.3	649.1
1.0	Ramp	0.000	-305	-23	-174.5	47.7	32732.0
1.0	Exp	0.000	-278	10	-193.3	43.4	39241.5
1.0	Sine	0.000	-324	-76	-242.7	34.8	60138.2
0.5	Step	0.698	-459	578	-143.4	140.1	40193.9
0.5	Ramp	0.100	-472	-103	-340.1	76.1	121447.4
0.5	Exp	0.000	-336	-135	-273.0	33.6	75686.6
0.5	Sine	0.000	-384	-183	-310.6	24.9	97091.6

Note: Negative Truncation points indicate transient observations are retained.

Table 17. AR(1) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = 0.3$							
2.0	Step	0.000	-10	699	27.0	97.7	10263.4
2.0	Ramp	0.003	-364	748	-18.4	115.4	13656.9
2.0	Exp	0.000	-388	732	25.8	119.8	15016.1
2.0	Sine	0.000	-450	684	-8.1	121.3	14778.8
1.5	Step	0.000	-21	731	29.8	106.2	12158.9
1.5	Ramp	0.003	-170	723	-33.6	121.3	15832.9
1.5	Exp	0.000	-420	737	-8.0	131.7	17400.8
1.5	Sine	0.000	-424	725	-61.8	153.1	27258.2
1.0	Step	0.008	-48	717	24.8	100.2	10657.5
1.0	Ramp	0.002	-223	721	-58.7	138.5	22611.5
1.0	Exp	0.000	-234	744	-79.8	148.2	28333.7
1.0	Sine	0.000	-454	740	-138.0	162.3	45400.2
0.5	Step	0.035	-452	677	-20.7	142.9	20841.2
0.5	Ramp	0.002	-406	744	-147.1	176.7	52857.7
0.5	Exp	0.000	-417	738	-167.2	172.8	57835.7
0.5	Sine	0.000	-460	734	-219.4	172.1	77772.8
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = -0.3$							
2.0	Step	0.004	-428	75	-5.7	32.4	1085.3
2.0	Ramp	0.000	-325	27	-92.2	44.4	10477.6
2.0	Exp	0.000	-341	54	-84.6	48.4	9503.5
2.0	Sine	0.000	-450	7	-145.2	64.3	25203.0
1.5	Step	0.005	-453	62	-9.0	46.5	2245.9
1.5	Ramp	0.000	-386	3	-121.7	46.9	16998.2
1.5	Exp	0.000	-382	26	-141.7	44.8	22075.0
1.5	Sine	0.001	-460	568	-201.5	60.1	44221.1
1.0	Step	0.021	-449	72	-24.2	60.4	4227.3
1.0	Ramp	0.001	-455	-26	-183.2	56.7	36781.9
1.0	Exp	0.000	-433	271	-207.6	44.7	45113.8
1.0	Sine	0.002	-468	312	-259.9	49.9	70035.5
0.5	Step	0.710	-457	41	-170.5	131.6	46381.0
0.5	Ramp	0.095	-471	-74	-342.7	73.5	122821.1
0.5	Exp	0.000	-458	-85	-274.0	40.8	76769.3
0.5	Sine	0.010	-474	-214	-319.3	34.1	103105.4

Note: Negative Truncation points indicate transient observations are retained.

Table 18. AR(1) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = 0.3$							
2.0	Step	0.000	-296	555	-35.4	77.9	7327.2
2.0	Ramp	0.000	-400	596	-85.8	106.3	18660.6
2.0	Exp	0.000	-343	524	-53.8	100.8	13066.6
2.0	Sine	0.008	-485	611	-105.9	149.3	33524.9
1.5	Step	0.000	-376	619	-49.3	110.0	14532.0
1.5	Ramp	0.000	-431	379	-102.3	96.1	19705.7
1.5	Exp	0.000	-393	605	-85.2	109.7	19288.8
1.5	Sine	0.005	-484	564	-146.7	151.5	44481.8
1.0	Step	0.004	-467	635	-62.0	111.9	16368.4
1.0	Ramp	0.003	-484	587	-137.2	119.4	33064.4
1.0	Exp	0.000	-447	639	-133.0	106.8	29082.4
1.0	Sine	0.005	-487	658	-217.0	143.3	67606.7
0.5	Step	0.043	-464	594	-76.9	130.6	22989.0
0.5	Ramp	0.008	-482	504	-201.4	113.0	53349.3
0.5	Exp	0.000	-488	624	-225.4	99.8	60766.1
0.5	Sine	0.003	-487	659	-286.9	111.5	94730.5
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = -0.3$							
2.0	Step	0.070	-394	61	-50.5	57.8	5894.7
2.0	Ramp	0.039	-328	16	-155.1	72.9	29382.3
2.0	Exp	0.016	-314	5	-141.2	71.0	24970.2
2.0	Sine	0.046	-481	-19	-298.7	124.5	104721.8
1.5	Step	0.036	-452	36	-102.7	111.0	22867.4
1.5	Ramp	0.019	-461	-7	-188.4	83.9	42526.5
1.5	Exp	0.016	-380	2	-195.6	67.3	42787.9
1.5	Sine	0.111	-482	-72	-328.9	112.3	120818.4
1.0	Step	0.171	-470	40	-123.3	119.9	29594.1
1.0	Ramp	0.006	-478	-51	-252.3	88.2	71431.2
1.0	Exp	0.027	-452	-25	-249.4	63.1	66169.8
1.0	Sine	0.186	-483	-129	-349.2	88.9	129863.5
0.5	Step	0.886	-450	33	-222.3	135.4	67742.6
0.5	Ramp	0.232	-476	-99	-371.4	66.3	142324.7
0.5	Exp	0.032	-489	-113	-310.9	53.7	99554.0
0.5	Sine	0.264	-485	-233	-354.2	59.8	129029.3

Note: Negative Truncation points indicate transient observations are retained.

Table 19. AR(1) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = -1.0$ and $\phi_2 = -0.5$							
2.0	Step	0.011	-470	28	-145.4	124.3	36588.6
2.0	Ramp	0.001	-486	-14	-229.0	118.9	66576.0
2.0	Exp	0.000	-486	13	-187.6	114.5	48295.3
2.0	Sine	0.114	-483	4	-281.9	142.3	99683.5
1.5	Step	0.029	-468	27	-155.7	125.3	39934.8
1.5	Ramp	0.002	-483	-38	-237.1	101.7	66534.9
1.5	Exp	0.000	-483	0	-228.5	102.6	62758.4
1.5	Sine	0.131	-483	-72	-327.6	115.9	120766.1
1.0	Step	0.232	-464	18	-149.8	112.8	35170.2
1.0	Ramp	0.004	-483	-63	-278.9	87.5	85432.5
1.0	Exp	0.004	-491	-86	-279.3	79.1	84258.2
1.0	Sine	0.170	-485	-181	-362.9	92.9	140314.8
0.5	Step	0.968	-437	3	-281.5	114.3	92319.4
0.5	Ramp	0.252	-470	-154	-392.5	50.2	156579.5
0.5	Exp	0.009	-494	-146	-344.1	61.0	122093.4
0.5	Sine	0.188	-485	-246	-384.6	69.2	152685.7
AR(2) Data with $\phi_1 = -1.5$ and $\phi_2 = -0.8$							
2.0	Step	0.000	-378	3	-148.8	104.0	32968.2
2.0	Ramp	0.000	-440	-11	-211.7	104.6	55736.8
2.0	Exp	0.000	-407	4	-158.6	111.9	37671.1
2.0	Sine	0.008	-466	2	-173.6	136.5	48783.4
1.5	Step	0.044	-458	4	-158.7	116.5	38753.3
1.5	Ramp	0.011	-487	-42	-229.2	102.7	63090.8
1.5	Exp	0.004	-481	3	-182.7	122.5	48398.5
1.5	Sine	0.029	-480	-3	-248.1	146.6	83036.6
1.0	Step	0.189	-459	10	-147.3	109.8	33749.1
1.0	Ramp	0.013	-486	-120	-264.7	90.3	78208.1
1.0	Exp	0.010	-490	-93	-250.4	89.1	70626.8
1.0	Sine	0.121	-457	-186	-338.5	100.7	124757.0
0.5	Step	0.997	-301	-119	-210.0	91.0	52381.0
0.5	Ramp	0.071	-472	-223	-389.4	41.6	153351.1
0.5	Exp	0.033	-491	-161	-329.8	54.7	111746.6
0.5	Sine	0.130	-480	-311	-394.1	60.7	159022.1

Note: Negative Truncation points indicate transient observations are retained.

Table 20. AR(2) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = 1.5$ and $\phi_2 = -0.8$							
2.0	Step	0.000	-1	742	30.3	101.9	11300.6
2.0	Ramp	0.000	-49	738	17.6	116.2	13815.6
2.0	Exp	0.000	-1	722	33.4	113.2	13941.5
2.0	Sine	0.000	-24	737	29.5	108.4	12625.3
1.5	Step	0.000	-1	732	27.9	97.7	10317.1
1.5	Ramp	0.000	-60	692	12.2	122.6	15177.5
1.5	Exp	0.000	-4	739	31.7	112.7	13718.3
1.5	Sine	0.000	-37	725	30.1	113.3	13730.3
1.0	Step	0.000	-2	735	31.0	110.9	13257.8
1.0	Ramp	0.000	-89	728	1.2	125.4	15724.4
1.0	Exp	0.000	-32	739	33.5	117.5	14938.2
1.0	Sine	0.000	-436	727	25.7	123.6	15926.0
0.5	Step	0.000	-53	712	35.8	116.6	14868.9
0.5	Ramp	0.000	-180	735	-43.6	125.0	17519.7
0.5	Exp	0.000	-185	741	-46.5	144.1	22910.9
0.5	Sine	0.000	-437	707	-92.6	154.2	32356.8
AR(2) Data with $\phi_1 = 1.0$ and $\phi_2 = -0.5$							
2.0	Step	0.000	-57	667	16.3	67.6	4835.9
2.0	Ramp	0.000	-194	669	-23.1	83.2	7457.8
2.0	Exp	0.000	-153	691	13.3	68.2	4827.2
2.0	Sine	0.000	-345	633	3.2	74.7	5591.8
1.5	Step	0.000	-52	714	15.9	69.6	5095.7
1.5	Ramp	0.000	-184	561	-40.3	67.8	6222.4
1.5	Exp	0.000	-169	732	-4.4	88.7	7886.3
1.5	Sine	0.000	-317	706	-44.1	86.5	9430.4
1.0	Step	0.000	-136	718	17.6	81.2	6900.1
1.0	Ramp	0.000	-227	721	-62.6	96.1	13159.7
1.0	Exp	0.000	-286	718	-77.4	94.4	14896.0
1.0	Sine	0.000	-439	654	-126.8	108.1	27770.3
0.5	Step	0.001	-420	651	3.0	80.2	6434.2
0.5	Ramp	0.000	-333	738	-140.7	106.1	31054.2
0.5	Exp	0.000	-305	725	-177.6	113.6	44458.7
0.5	Sine	0.000	-438	725	-220.8	128.1	65173.0

Note: Negative Truncation points indicate transient observations are retained.

Table 21. AR(2) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = 0.3$							
2.0	Step	0.000	-221	729	35.4	118.1	15207.2
2.0	Ramp	0.000	-247	732	-3.5	134.5	18095.7
2.0	Exp	0.000	-317	745	29.9	122.2	15825.8
2.0	Sine	0.001	-441	739	11.0	124.6	15657.7
1.5	Step	0.002	-64	716	32.9	110.4	13265.0
1.5	Ramp	0.000	-247	746	-25.7	124.5	16159.8
1.5	Exp	0.000	-297	737	4.8	137.4	18900.9
1.5	Sine	0.000	-470	736	-38.1	153.0	24847.6
1.0	Step	0.006	-251	735	28.7	108.5	12603.3
1.0	Ramp	0.000	-299	721	-46.0	140.2	21779.3
1.0	Exp	0.000	-318	733	-65.5	150.6	26971.5
1.0	Sine	0.003	-471	740	-110.2	173.1	42107.5
0.5	Step	0.024	-458	742	7.0	122.7	15112.9
0.5	Ramp	0.004	-467	735	-129.4	174.9	47331.8
0.5	Exp	0.000	-374	739	-159.1	166.9	53170.0
0.5	Sine	0.001	-447	741	-203.3	177.8	72951.9
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = -0.3$							
2.0	Step	0.000	-120	593	0.1	47.0	2213.2
2.0	Ramp	0.000	-263	596	-61.6	64.8	7989.9
2.0	Exp	0.000	-277	543	-34.3	65.4	5450.6
2.0	Sine	0.000	-482	624	-94.0	100.9	19027.8
1.5	Step	0.000	-243	600	-3.5	53.1	2828.6
1.5	Ramp	0.000	-328	437	-81.1	57.2	9857.3
1.5	Exp	0.000	-294	595	-94.0	59.9	12418.7
1.5	Sine	0.001	-475	577	-157.2	98.7	34446.5
1.0	Step	0.001	-440	647	-11.5	76.4	5968.9
1.0	Ramp	0.000	-393	603	-125.6	82.8	22631.2
1.0	Exp	0.000	-368	639	-162.7	68.2	31133.6
1.0	Sine	0.006	-473	591	-220.8	84.8	55966.4
0.5	Step	0.100	-470	637	-83.0	120.8	21485.2
0.5	Ramp	0.001	-470	636	-234.0	96.4	64021.3
0.5	Exp	0.000	-417	647	-237.6	71.6	61571.1
0.5	Sine	0.017	-478	624	-288.3	79.9	89471.6

Note: Negative Truncation points indicate transient observations are retained.

Table 22. AR(2) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = 0.3$							
2.0	Step	0.000	-174	586	-20.7	51.9	3122.5
2.0	Ramp	0.000	-290	497	-81.9	80.2	13154.1
2.0	Exp	0.000	-261	164	-48.2	72.1	7528.1
2.0	Sine	0.003	-482	355	-117.6	141.8	33943.3
1.5	Step	0.005	-357	610	-30.4	70.0	5832.0
1.5	Ramp	0.000	-330	523	-103.7	82.8	17609.2
1.5	Exp	0.000	-333	567	-81.0	84.3	13662.8
1.5	Sine	0.004	-484	326	-171.2	138.2	48401.6
1.0	Step	0.031	-458	646	-45.4	91.6	10457.3
1.0	Ramp	0.000	-420	494	-141.3	95.0	28991.1
1.0	Exp	0.000	-373	639	-148.6	85.2	29340.4
1.0	Sine	0.006	-482	658	-237.4	123.5	71598.3
0.5	Step	0.140	-472	594	-79.6	115.5	19678.9
0.5	Ramp	0.011	-472	407	-217.1	92.7	55729.9
0.5	Exp	0.000	-439	288	-240.4	70.6	62782.2
0.5	Sine	0.008	-486	-74	-311.5	85.0	104283.0
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = -0.3$							
2.0	Step	0.000	-325	80	-86.0	89.9	15482.7
2.0	Ramp	0.000	-436	59	-173.6	105.7	41293.3
2.0	Exp	0.000	-380	62	-144.1	103.5	31484.2
2.0	Sine	0.050	-485	22	-268.4	151.3	94918.7
1.5	Step	0.026	-468	51	-116.9	122.3	28613.1
1.5	Ramp	0.003	-481	-10	-201.9	116.7	54370.5
1.5	Exp	0.000	-436	43	-199.2	97.7	49217.0
1.5	Sine	0.058	-484	-7	-295.0	130.9	104164.6
1.0	Step	0.097	-471	66	-119.5	125.5	30018.6
1.0	Ramp	0.008	-480	-18	-239.4	106.9	68715.6
1.0	Exp	0.000	-456	-39	-248.3	87.6	69338.8
1.0	Sine	0.045	-485	447	-328.0	110.6	119831.2
0.5	Step	0.475	-457	44	-195.1	131.4	55317.7
0.5	Ramp	0.051	-477	108	-320.0	87.4	110053.6
0.5	Exp	0.000	-490	-70	-320.8	77.9	108992.6
0.5	Sine	0.071	-486	-187	-358.1	81.1	134784.4

Note: Negative Truncation points indicate transient observations are retained.

Table 23. AR(2) with Measurement Noise Model Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = -1.0$ and $\phi_2 = -0.5$							
2.0	Step	0.000	-334	56	-123.1	91.3	23477.2
2.0	Ramp	0.000	-418	-5	-210.3	104.2	55083.9
2.0	Exp	0.000	-381	32	-169.3	106.1	39930.1
2.0	Sine	0.124	-482	40	-269.0	159.5	97796.9
1.5	Step	0.021	-471	41	-167.4	130.9	45163.6
1.5	Ramp	0.003	-483	-5	-240.4	113.4	70675.7
1.5	Exp	0.000	-432	26	-206.1	107.0	53938.8
1.5	Sine	0.112	-484	0	-312.4	137.6	116507.5
1.0	Step	0.055	-466	40	-181.8	135.9	51503.5
1.0	Ramp	0.008	-482	-26	-270.4	106.4	84439.8
1.0	Exp	0.000	-475	-18	-271.2	97.1	82987.0
1.0	Sine	0.197	-485	-51	-343.3	117.0	131582.3
0.5	Step	0.373	-473	17	-224.6	129.8	67326.8
0.5	Ramp	0.019	-478	-100	-344.0	76.9	124233.9
0.5	Exp	0.006	-493	-91	-341.6	75.9	122431.6
0.5	Sine	0.194	-485	-192	-371.1	86.8	145279.2
AR(2) Data with $\phi_1 = -1.5$ and $\phi_2 = -0.8$							
2.0	Step	0.000	-185	28	-64.3	41.2	5829.0
2.0	Ramp	0.000	-324	19	-163.3	67.5	31218.6
2.0	Exp	0.000	-254	13	-99.4	58.1	13261.1
2.0	Sine	0.004	-477	22	-199.8	142.9	60336.3
1.5	Step	0.000	-270	35	-100.0	64.3	14134.6
1.5	Ramp	0.000	-390	1	-200.0	80.5	46495.3
1.5	Exp	0.000	-351	19	-153.8	90.2	31787.4
1.5	Sine	0.028	-480	7	-242.8	154.3	82746.1
1.0	Step	0.003	-473	19	-188.4	129.1	52176.8
1.0	Ramp	0.000	-487	-9	-263.1	104.3	80098.6
1.0	Exp	0.000	-440	7	-209.8	100.7	54174.3
1.0	Sine	0.143	-483	1	-296.2	155.9	112013.6
0.5	Step	0.100	-468	29	-166.9	87.5	35530.1
0.5	Ramp	0.039	-487	-52	-293.3	71.2	91118.5
0.5	Exp	0.007	-494	-83	-313.7	78.9	104660.1
0.5	Sine	0.124	-486	-131	-369.1	97.8	145761.4

Note: Negative Truncation points indicate transient observations are retained.

For the confidence interval technique of NOBM, Fishman's algorithm [22, 23] to determine the appropriate batch size is applied. (See discussion on page 27.) In the following tables, the column labeled 'Fail' indicates the frequency that Fishman's algorithm did not find a batch size which results in statistically independent batches. The OBM technique is applied with the same batch size as the nonoverlapping batch means technique. Therefore, if Fishman's algorithm failed, neither confidence interval technique is applied to the sequence. The number and size of the batches selected by Fishman's algorithm depended upon the type of variation in the data.

The confidence interval and bias evaluations for the truncated sequence are presented in Tables 24, 25, 26 and 27 for the AR(1) with measurement noise model and Tables 28, 29, 30 and 31 for the AR(2) with measurement noise formulation. Sequences without a transient are generated by truncating at the known end of transient, n_0 . The results for these sequences are shown in the rows are labeled "None". Nonoverlapping and Overlapping Batch Means produced practically identical coverage rates, half widths, and half width variance for the sequences without a transient. Coverage rates are generally near the nominal rate of 0.9 for the sequences without a transient.

On the average, the estimated truncation point \hat{n}_0 includes some transient data in the truncated data set. Therefore, the actual coverage rate for the sequences truncated at the estimated truncation point \hat{n}_0 decreases from the data set truncated at the true truncation point n_0 . Even though the estimated truncation point for the exponentially decaying sine wave transient generally includes the most transient values, the corresponding coverage rates do not decrease because high and low transient values apparently cancel each other out. For the "Sine" sequences, the confidence interval half widths also increase marginally in magnitude and variance.

Schruben's initial bias test, Equation (10) on page 29, requires an estimate of the variance of the sample mean. The modified version of Fishman's formula to approximate this variance for an autoregressive sequence, Equation 56, is applied. When the population parameters that generated the sequence are used in the Equation (11), the variance $\sigma_{\mu_y}^2$ is

Table 24. AR(1) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient $\Delta\mu_y(\sigma_y)$ Type		Fail	NOBM Cov Half Width		OBM Cov Half Width		Bias Test $F(\hat{\sigma}_y^2)$ $\chi^2(\sigma_y^2)$	
AR(2) Data with $\phi_1 = 1.5$ and $\phi_2 = -0.8$								
None		0.01	0.93	0.194 ± 0.040	0.91	0.197 ± 0.082	1.00	0.89
2.0	Step	0.00	0.93	0.198 ± 0.053	0.93	0.200 ± 0.050	1.00	0.89
2.0	Ramp	0.02	0.85	0.224 ± 0.090	0.85	0.225 ± 0.071	0.85	0.34
2.0	Exp	0.01	0.92	0.198 ± 0.057	0.92	0.200 ± 0.052	1.00	0.90
2.0	Sine	0.01	0.92	0.209 ± 0.199	0.92	0.211 ± 0.202	1.00	0.87
1.5	Step	0.01	0.92	0.197 ± 0.059	0.93	0.200 ± 0.056	1.00	0.91
1.5	Ramp	0.02	0.75	0.239 ± 0.152	0.75	0.238 ± 0.124	0.70	0.24
1.5	Exp	0.02	0.90	0.206 ± 0.062	0.90	0.207 ± 0.068	0.95	0.75
1.5	Sine	0.01	0.84	0.222 ± 0.155	0.85	0.227 ± 0.162	0.98	0.87
1.0	Step	0.02	0.92	0.199 ± 0.063	0.92	0.201 ± 0.064	0.98	0.88
1.0	Ramp	0.03	0.56	0.254 ± 0.144	0.56	0.255 ± 0.148	0.46	0.14
1.0	Exp	0.09	0.50	0.259 ± 0.138	0.50	0.259 ± 0.122	0.34	0.17
1.0	Sine	0.00	0.91	0.268 ± 0.103	0.92	0.278 ± 0.112	0.67	0.56
0.5	Step	0.26	0.76	0.238 ± 0.105	0.76	0.238 ± 0.097	0.51	0.33
0.5	Ramp	0.11	0.30	0.249 ± 0.072	0.31	0.252 ± 0.094	0.25	0.09
0.5	Exp	0.05	0.22	0.243 ± 0.059	0.23	0.246 ± 0.075	0.22	0.07
0.5	Sine	0.00	0.92	0.240 ± 0.063	0.93	0.243 ± 0.070	0.83	0.54
AR(2) Data with $\phi_1 = 1.0$ and $\phi_2 = -0.5$								
None		0.00	0.91	0.109 ± 0.014	0.91	0.117 ± 0.050	1.00	0.90
2.0	Step	0.00	0.92	0.110 ± 0.014	0.92	0.110 ± 0.013	1.00	0.91
2.0	Ramp	0.03	0.70	0.118 ± 0.041	0.70	0.118 ± 0.035	0.56	0.25
2.0	Exp	0.10	0.73	0.118 ± 0.022	0.76	0.124 ± 0.042	0.54	0.31
2.0	Sine	0.03	0.81	0.127 ± 0.046	0.83	0.137 ± 0.057	0.59	0.55
1.5	Step	0.00	0.91	0.109 ± 0.014	0.91	0.109 ± 0.013	1.00	0.92
1.5	Ramp	0.04	0.63	0.140 ± 0.118	0.61	0.132 ± 0.077	0.36	0.14
1.5	Exp	0.22	0.50	0.148 ± 0.137	0.46	0.140 ± 0.096	0.18	0.07
1.5	Sine	0.00	0.85	0.130 ± 0.049	0.87	0.136 ± 0.052	0.74	0.63
1.0	Step	0.03	0.87	0.113 ± 0.023	0.88	0.114 ± 0.030	0.87	0.73
1.0	Ramp	0.07	0.47	0.149 ± 0.104	0.51	0.155 ± 0.124	0.17	0.05
1.0	Exp	0.08	0.32	0.137 ± 0.082	0.36	0.142 ± 0.100	0.11	0.03
1.0	Sine	0.00	0.89	0.120 ± 0.036	0.89	0.124 ± 0.040	0.45	0.30
0.5	Step	0.30	0.65	0.126 ± 0.082	0.64	0.126 ± 0.078	0.31	0.20
0.5	Ramp	0.52	0.25	0.117 ± 0.055	0.26	0.119 ± 0.063	0.03	0.01
0.5	Exp	0.07	0.35	0.112 ± 0.042	0.34	0.111 ± 0.029	0.23	0.03
0.5	Sine	0.00	0.86	0.107 ± 0.017	0.87	0.109 ± 0.020	0.94	0.85

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 25. AR(1) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient $\Delta\mu_y(\sigma_y)$	Type	Fail	NOBM		OBM		Bias Test	
			Cov	Half Width	Cov	Half Width	$F(\hat{\sigma}_y^2)$	$\chi^2(\sigma_y^2)$
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = 0.3$								
None		0.02	0.86	0.240 ± 0.072	0.89	0.281 ± 0.122	0.90	0.94
2.0	Step	0.03	0.85	0.241 ± 0.073	0.84	0.232 ± 0.064	0.91	0.94
2.0	Ramp	0.05	0.82	0.244 ± 0.094	0.81	0.240 ± 0.105	0.79	0.76
2.0	Exp	0.03	0.83	0.241 ± 0.078	0.82	0.234 ± 0.074	0.88	0.88
2.0	Sine	0.03	0.82	0.261 ± 0.145	0.82	0.265 ± 0.166	0.87	0.91
1.5	Step	0.03	0.83	0.236 ± 0.072	0.82	0.229 ± 0.064	0.95	0.95
1.5	Ramp	0.05	0.79	0.251 ± 0.107	0.79	0.246 ± 0.092	0.72	0.66
1.5	Exp	0.13	0.78	0.253 ± 0.110	0.77	0.250 ± 0.104	0.66	0.63
1.5	Sine	0.04	0.84	0.276 ± 0.151	0.86	0.283 ± 0.171	0.74	0.71
1.0	Step	0.02	0.83	0.240 ± 0.079	0.82	0.231 ± 0.070	0.92	0.94
1.0	Ramp	0.07	0.81	0.275 ± 0.170	0.79	0.263 ± 0.125	0.55	0.56
1.0	Exp	0.14	0.76	0.300 ± 0.190	0.74	0.284 ± 0.145	0.38	0.40
1.0	Sine	0.02	0.86	0.267 ± 0.129	0.87	0.274 ± 0.150	0.80	0.81
0.5	Step	0.15	0.83	0.270 ± 0.200	0.82	0.262 ± 0.177	0.65	0.66
0.5	Ramp	0.16	0.72	0.284 ± 0.152	0.75	0.292 ± 0.168	0.32	0.36
0.5	Exp	0.06	0.72	0.270 ± 0.124	0.72	0.271 ± 0.128	0.40	0.39
0.5	Sine	0.01	0.84	0.242 ± 0.089	0.85	0.247 ± 0.101	0.80	0.78
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = -0.3$								
None		0.00	0.92	0.069 ± 0.004	0.91	0.073 ± 0.031	0.99	0.89
2.0	Step	0.06	0.89	0.072 ± 0.018	0.90	0.073 ± 0.028	0.88	0.80
2.0	Ramp	0.02	0.73	0.074 ± 0.045	0.73	0.074 ± 0.050	0.45	0.27
2.0	Exp	0.10	0.61	0.076 ± 0.049	0.61	0.076 ± 0.044	0.27	0.15
2.0	Sine	0.00	0.85	0.075 ± 0.043	0.88	0.080 ± 0.052	0.55	0.44
1.5	Step	0.05	0.89	0.071 ± 0.014	0.90	0.073 ± 0.027	0.88	0.79
1.5	Ramp	0.03	0.64	0.074 ± 0.041	0.63	0.073 ± 0.034	0.34	0.19
1.5	Exp	0.05	0.49	0.074 ± 0.042	0.48	0.073 ± 0.033	0.13	0.06
1.5	Sine	0.00	0.83	0.072 ± 0.034	0.85	0.075 ± 0.045	0.77	0.71
1.0	Step	0.08	0.85	0.072 ± 0.025	0.85	0.072 ± 0.025	0.75	0.63
1.0	Ramp	0.06	0.49	0.074 ± 0.033	0.52	0.076 ± 0.046	0.18	0.09
1.0	Exp	0.03	0.41	0.068 ± 0.021	0.42	0.069 ± 0.027	0.15	0.05
1.0	Sine	0.00	0.86	0.068 ± 0.026	0.88	0.072 ± 0.041	0.50	0.33
0.5	Step	0.19	0.60	0.073 ± 0.038	0.61	0.072 ± 0.036	0.25	0.16
0.5	Ramp	0.27	0.25	0.066 ± 0.025	0.25	0.066 ± 0.027	0.06	0.02
0.5	Exp	0.03	0.55	0.064 ± 0.014	0.55	0.065 ± 0.021	0.29	0.10
0.5	Sine	0.00	0.87	0.065 ± 0.026	0.88	0.067 ± 0.031	0.92	0.88

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 26. AR(1) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient $\Delta\mu_y(\sigma_y)$ Type		Fail	NOBM Cov Half Width		OBM Cov Half Width		Bias Test $F(\hat{\sigma}_y^2)$ $\chi^2(\sigma_y^2)$	
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = 0.3$								
None		0.00	0.90	0.044 ± 0.005	0.91	0.049 ± 0.021	0.85	0.88
2.0	Step	0.27	0.80	0.053 ± 0.034	0.84	0.055 ± 0.053	0.47	0.53
2.0	Ramp	0.13	0.79	0.064 ± 0.076	0.81	0.067 ± 0.094	0.38	0.46
2.0	Exp	0.23	0.81	0.056 ± 0.072	0.84	0.057 ± 0.067	0.43	0.50
2.0	Sine	0.00	0.86	0.069 ± 0.089	0.91	0.077 ± 0.092	0.63	0.70
1.5	Step	0.31	0.85	0.051 ± 0.032	0.87	0.053 ± 0.045	0.47	0.50
1.5	Ramp	0.12	0.79	0.059 ± 0.056	0.79	0.059 ± 0.063	0.34	0.43
1.5	Exp	0.23	0.77	0.054 ± 0.047	0.77	0.054 ± 0.051	0.31	0.38
1.5	Sine	0.00	0.82	0.060 ± 0.066	0.88	0.071 ± 0.076	0.61	0.65
1.0	Step	0.29	0.83	0.050 ± 0.030	0.86	0.052 ± 0.036	0.44	0.48
1.0	Ramp	0.17	0.77	0.052 ± 0.033	0.76	0.051 ± 0.035	0.23	0.33
1.0	Exp	0.12	0.69	0.048 ± 0.023	0.71	0.048 ± 0.027	0.21	0.29
1.0	Sine	0.00	0.83	0.057 ± 0.063	0.90	0.069 ± 0.074	0.56	0.63
0.5	Step	0.26	0.85	0.050 ± 0.032	0.84	0.050 ± 0.028	0.40	0.45
0.5	Ramp	0.17	0.69	0.046 ± 0.018	0.70	0.046 ± 0.020	0.16	0.23
0.5	Exp	0.12	0.69	0.045 ± 0.019	0.69	0.045 ± 0.027	0.15	0.25
0.5	Sine	0.00	0.86	0.052 ± 0.051	0.91	0.058 ± 0.054	0.48	0.57
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = -0.3$								
None		0.00	0.91	0.032 ± 0.005	0.89	0.034 ± 0.014	0.99	0.86
2.0	Step	0.15	0.58	0.044 ± 0.019	0.64	0.047 ± 0.029	0.36	0.30
2.0	Ramp	0.07	0.49	0.045 ± 0.033	0.49	0.045 ± 0.035	0.17	0.08
2.0	Exp	0.10	0.39	0.042 ± 0.028	0.39	0.042 ± 0.027	0.10	0.04
2.0	Sine	0.00	0.85	0.062 ± 0.059	0.92	0.071 ± 0.060	0.75	0.46
1.5	Step	0.19	0.46	0.042 ± 0.016	0.54	0.045 ± 0.023	0.28	0.23
1.5	Ramp	0.10	0.42	0.040 ± 0.019	0.44	0.040 ± 0.023	0.13	0.06
1.5	Exp	0.10	0.30	0.038 ± 0.014	0.31	0.038 ± 0.015	0.05	0.02
1.5	Sine	0.00	0.81	0.057 ± 0.055	0.93	0.073 ± 0.064	0.83	0.61
1.0	Step	0.33	0.57	0.038 ± 0.022	0.57	0.039 ± 0.021	0.28	0.20
1.0	Ramp	0.18	0.30	0.036 ± 0.009	0.30	0.037 ± 0.012	0.05	0.02
1.0	Exp	0.09	0.31	0.036 ± 0.011	0.32	0.036 ± 0.014	0.07	0.03
1.0	Sine	0.00	0.82	0.046 ± 0.037	0.95	0.060 ± 0.048	0.65	0.45
0.5	Step	0.03	0.30	0.035 ± 0.008	0.30	0.035 ± 0.007	0.16	0.10
0.5	Ramp	0.11	0.14	0.033 ± 0.005	0.15	0.034 ± 0.006	0.02	0.01
0.5	Exp	0.07	0.40	0.034 ± 0.009	0.42	0.035 ± 0.016	0.13	0.03
0.5	Sine	0.00	0.88	0.036 ± 0.017	0.92	0.039 ± 0.022	0.85	0.77

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 27. AR(1) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient		Fail	NOBM		OBM		Bias Test	
$\Delta\mu_y(\sigma_y)$	Type		Cov	Half Width	Cov	Half Width	$F(\hat{\sigma}_y^2)$	$\chi^2(\sigma_y^2)$
AR(2) Data with $\phi_1 = -1.0$ and $\phi_2 = -0.5$								
None		0.01	0.94	0.025 ± 0.006	0.91	0.024 ± 0.010	1.00	0.82
2.0	Step	0.55	0.40	0.043 ± 0.018	0.39	0.042 ± 0.017	0.11	0.09
2.0	Ramp	0.38	0.53	0.054 ± 0.044	0.49	0.048 ± 0.032	0.08	0.02
2.0	Exp	0.53	0.34	0.043 ± 0.033	0.33	0.042 ± 0.024	0.06	0.02
2.0	Sine	0.01	0.93	0.068 ± 0.067	0.95	0.071 ± 0.067	0.65	0.52
1.5	Step	0.27	0.26	0.044 ± 0.013	0.30	0.045 ± 0.015	0.10	0.08
1.5	Ramp	0.22	0.40	0.049 ± 0.031	0.42	0.049 ± 0.032	0.04	0.01
1.5	Exp	0.39	0.31	0.046 ± 0.032	0.29	0.044 ± 0.029	0.01	0.00
1.5	Sine	0.00	0.89	0.058 ± 0.056	0.97	0.065 ± 0.055	0.66	0.48
1.0	Step	0.43	0.35	0.038 ± 0.017	0.40	0.039 ± 0.016	0.11	0.08
1.0	Ramp	0.24	0.20	0.040 ± 0.016	0.32	0.046 ± 0.028	0.01	0.00
1.0	Exp	0.34	0.18	0.035 ± 0.013	0.23	0.038 ± 0.021	0.00	0.00
1.0	Sine	0.00	0.84	0.044 ± 0.040	0.96	0.054 ± 0.041	0.55	0.43
0.5	Step	0.40	0.27	0.034 ± 0.019	0.27	0.034 ± 0.018	0.04	0.04
0.5	Ramp	0.55	0.07	0.030 ± 0.008	0.08	0.030 ± 0.010	0.00	0.00
0.5	Exp	0.39	0.23	0.033 ± 0.013	0.27	0.035 ± 0.021	0.03	0.01
0.5	Sine	0.00	0.96	0.039 ± 0.031	0.97	0.039 ± 0.026	0.73	0.58
AR(2) Data with $\phi_1 = -1.5$ and $\phi_2 = -0.8$								
None		0.09	0.98	0.024 ± 0.007	0.96	0.021 ± 0.007	0.99	0.58
2.0	Step	0.61	0.39	0.074 ± 0.030	0.31	0.071 ± 0.032	0.04	0.02
2.0	Ramp	0.30	0.33	0.070 ± 0.061	0.41	0.073 ± 0.073	0.04	0.00
2.0	Exp	0.46	0.31	0.062 ± 0.027	0.43	0.066 ± 0.028	0.04	0.02
2.0	Sine	0.02	0.90	0.076 ± 0.064	0.94	0.078 ± 0.063	0.70	0.30
1.5	Step	0.39	0.28	0.068 ± 0.024	0.34	0.071 ± 0.026	0.03	0.02
1.5	Ramp	0.23	0.22	0.059 ± 0.038	0.30	0.067 ± 0.065	0.02	0.00
1.5	Exp	0.38	0.29	0.055 ± 0.023	0.29	0.055 ± 0.023	0.03	0.01
1.5	Sine	0.00	0.97	0.069 ± 0.046	0.98	0.070 ± 0.045	0.74	0.29
1.0	Step	0.48	0.42	0.051 ± 0.016	0.48	0.053 ± 0.018	0.05	0.03
1.0	Ramp	0.30	0.15	0.051 ± 0.018	0.15	0.051 ± 0.018	0.00	0.00
1.0	Exp	0.29	0.06	0.052 ± 0.018	0.06	0.051 ± 0.015	0.00	0.00
1.0	Sine	0.00	0.99	0.064 ± 0.036	1.00	0.065 ± 0.037	0.64	0.27
0.5	Step	0.50	0.00	0.045 ± 0.000	0.00	0.046 ± 0.000	0.00	0.00
0.5	Ramp	0.14	0.01	0.045 ± 0.005	0.00	0.045 ± 0.004	0.00	0.00
0.5	Exp	0.31	0.04	0.042 ± 0.011	0.05	0.042 ± 0.010	0.00	0.00
0.5	Sine	0.00	0.93	0.045 ± 0.027	0.99	0.051 ± 0.023	0.77	0.38

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 28. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient $\Delta\mu_{\mathbf{v}}(\sigma_{\mathbf{v}})$ Type		Fail	NOBM Cov Half Width		OBM Cov Half Width		Bias Test $F(\hat{\sigma}_{\mathbf{v}}^2)$ $\chi^2(\sigma_{\mathbf{v}}^2)$	
AR(2) Data with $\phi_1 = 1.5$ and $\phi_2 = -0.8$								
None		0.00	0.93	0.194 ± 0.039	0.93	0.196 ± 0.036	0.78	0.88
2.0	Step	0.00	0.92	0.198 ± 0.050	0.93	0.200 ± 0.049	0.83	0.91
2.0	Ramp	0.01	0.92	0.198 ± 0.057	0.92	0.199 ± 0.053	0.75	0.86
2.0	Exp	0.01	0.91	0.197 ± 0.055	0.92	0.200 ± 0.054	0.83	0.92
2.0	Sine	0.01	0.91	0.198 ± 0.055	0.91	0.200 ± 0.052	0.75	0.86
1.5	Step	0.01	0.91	0.196 ± 0.052	0.92	0.198 ± 0.047	0.84	0.92
1.5	Ramp	0.00	0.91	0.197 ± 0.056	0.91	0.200 ± 0.053	0.73	0.84
1.5	Exp	0.00	0.90	0.198 ± 0.057	0.91	0.199 ± 0.055	0.82	0.91
1.5	Sine	0.01	0.92	0.198 ± 0.054	0.92	0.200 ± 0.052	0.76	0.87
1.0	Step	0.00	0.91	0.198 ± 0.059	0.91	0.199 ± 0.057	0.83	0.90
1.0	Ramp	0.01	0.91	0.197 ± 0.055	0.92	0.199 ± 0.051	0.67	0.80
1.0	Exp	0.01	0.93	0.200 ± 0.057	0.92	0.201 ± 0.054	0.82	0.91
1.0	Sine	0.00	0.92	0.200 ± 0.069	0.92	0.201 ± 0.069	0.80	0.90
0.5	Step	0.01	0.92	0.199 ± 0.058	0.92	0.200 ± 0.055	0.84	0.93
0.5	Ramp	0.00	0.89	0.196 ± 0.054	0.89	0.198 ± 0.048	0.48	0.63
0.5	Exp	0.00	0.83	0.200 ± 0.058	0.84	0.202 ± 0.054	0.34	0.46
0.5	Sine	0.00	0.91	0.199 ± 0.059	0.91	0.201 ± 0.057	0.69	0.79
AR(2) Data with $\phi_1 = 1.0$ and $\phi_2 = -0.5$								
None		0.00	0.92	0.109 ± 0.014	0.92	0.110 ± 0.013	0.85	0.89
2.0	Step	0.00	0.91	0.111 ± 0.019	0.92	0.111 ± 0.018	0.88	0.91
2.0	Ramp	0.01	0.89	0.109 ± 0.027	0.89	0.110 ± 0.019	0.70	0.77
2.0	Exp	0.00	0.90	0.110 ± 0.032	0.90	0.111 ± 0.023	0.84	0.90
2.0	Sine	0.00	0.89	0.111 ± 0.026	0.90	0.111 ± 0.025	0.83	0.89
1.5	Step	0.00	0.90	0.111 ± 0.021	0.91	0.111 ± 0.019	0.88	0.92
1.5	Ramp	0.00	0.86	0.109 ± 0.024	0.86	0.109 ± 0.019	0.62	0.70
1.5	Exp	0.00	0.85	0.111 ± 0.019	0.86	0.112 ± 0.018	0.64	0.71
1.5	Sine	0.00	0.86	0.110 ± 0.017	0.86	0.111 ± 0.018	0.79	0.83
1.0	Step	0.00	0.89	0.110 ± 0.019	0.90	0.111 ± 0.020	0.88	0.91
1.0	Ramp	0.00	0.84	0.109 ± 0.020	0.84	0.109 ± 0.020	0.48	0.56
1.0	Exp	0.00	0.77	0.109 ± 0.017	0.78	0.109 ± 0.016	0.33	0.43
1.0	Sine	0.00	0.89	0.109 ± 0.026	0.89	0.109 ± 0.031	0.72	0.79
0.5	Step	0.01	0.87	0.111 ± 0.021	0.87	0.111 ± 0.018	0.76	0.81
0.5	Ramp	0.00	0.75	0.106 ± 0.020	0.75	0.107 ± 0.019	0.29	0.37
0.5	Exp	0.00	0.73	0.104 ± 0.019	0.73	0.105 ± 0.017	0.26	0.35
0.5	Sine	0.00	0.88	0.103 ± 0.020	0.88	0.103 ± 0.021	0.57	0.65

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 29. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient		Fail	NOBM		OBM		Bias Test	
$\Delta\mu_v(\sigma_v)$	Type		Cov	Half Width	Cov	Half Width	$F(\hat{\sigma}_y^2)$	$\chi^2(\sigma_y^2)$
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = 0.3$								
None		0.03	0.86	0.240 ± 0.071	0.85	0.231 ± 0.063	0.89	0.94
2.0	Step	0.02	0.83	0.241 ± 0.076	0.82	0.233 ± 0.066	0.90	0.94
2.0	Ramp	0.03	0.83	0.243 ± 0.106	0.82	0.238 ± 0.107	0.78	0.82
2.0	Exp	0.03	0.83	0.240 ± 0.100	0.82	0.233 ± 0.085	0.90	0.92
2.0	Sine	0.03	0.83	0.248 ± 0.111	0.81	0.245 ± 0.123	0.90	0.94
1.5	Step	0.03	0.83	0.238 ± 0.077	0.82	0.231 ± 0.077	0.92	0.95
1.5	Ramp	0.03	0.81	0.245 ± 0.096	0.79	0.240 ± 0.082	0.72	0.74
1.5	Exp	0.06	0.77	0.254 ± 0.148	0.77	0.249 ± 0.143	0.69	0.74
1.5	Sine	0.05	0.82	0.261 ± 0.111	0.82	0.264 ± 0.133	0.79	0.80
1.0	Step	0.02	0.84	0.240 ± 0.078	0.82	0.232 ± 0.070	0.92	0.95
1.0	Ramp	0.07	0.81	0.255 ± 0.127	0.80	0.249 ± 0.114	0.58	0.64
1.0	Exp	0.12	0.77	0.277 ± 0.154	0.76	0.267 ± 0.126	0.42	0.49
1.0	Sine	0.01	0.84	0.255 ± 0.098	0.84	0.257 ± 0.113	0.82	0.85
0.5	Step	0.08	0.83	0.247 ± 0.112	0.81	0.240 ± 0.099	0.77	0.80
0.5	Ramp	0.12	0.75	0.271 ± 0.131	0.77	0.271 ± 0.140	0.34	0.44
0.5	Exp	0.03	0.73	0.265 ± 0.123	0.73	0.263 ± 0.118	0.37	0.44
0.5	Sine	0.01	0.84	0.235 ± 0.073	0.85	0.237 ± 0.085	0.72	0.79
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = -0.3$								
None		0.00	0.92	0.069 ± 0.004	0.92	0.069 ± 0.004	0.88	0.89
2.0	Step	0.05	0.87	0.074 ± 0.025	0.88	0.075 ± 0.036	0.76	0.78
2.0	Ramp	0.01	0.84	0.074 ± 0.048	0.84	0.074 ± 0.051	0.53	0.56
2.0	Exp	0.03	0.82	0.072 ± 0.035	0.82	0.072 ± 0.028	0.52	0.54
2.0	Sine	0.00	0.84	0.074 ± 0.048	0.86	0.079 ± 0.060	0.71	0.72
1.5	Step	0.06	0.88	0.072 ± 0.015	0.90	0.074 ± 0.032	0.77	0.78
1.5	Ramp	0.01	0.78	0.070 ± 0.026	0.79	0.070 ± 0.032	0.45	0.49
1.5	Exp	0.02	0.69	0.070 ± 0.023	0.69	0.070 ± 0.027	0.24	0.27
1.5	Sine	0.00	0.87	0.072 ± 0.043	0.89	0.077 ± 0.057	0.65	0.66
1.0	Step	0.07	0.88	0.071 ± 0.026	0.89	0.072 ± 0.025	0.74	0.75
1.0	Ramp	0.03	0.70	0.068 ± 0.015	0.71	0.069 ± 0.023	0.30	0.32
1.0	Exp	0.04	0.67	0.066 ± 0.007	0.67	0.066 ± 0.011	0.20	0.22
1.0	Sine	0.00	0.87	0.068 ± 0.034	0.88	0.072 ± 0.042	0.64	0.64
0.5	Step	0.03	0.69	0.069 ± 0.011	0.68	0.069 ± 0.011	0.39	0.42
0.5	Ramp	0.03	0.56	0.064 ± 0.010	0.56	0.064 ± 0.009	0.13	0.15
0.5	Exp	0.02	0.71	0.064 ± 0.014	0.71	0.065 ± 0.020	0.24	0.27
0.5	Sine	0.00	0.88	0.067 ± 0.042	0.89	0.068 ± 0.038	0.72	0.75

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 30. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient		Fail	NOBM		OBM		Bias Test	
$\Delta\mu_{\nu}(\sigma_{\nu})$	Type		Cov	Half Width	Cov	Half Width	$F(\hat{\sigma}_{\nu}^2)$	$\chi^2(\sigma_{\nu}^2)$
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = 0.3$								
None		0.00	0.90	0.044 ± 0.004	0.90	0.044 ± 0.004	0.90	0.88
2.0	Step	0.20	0.78	0.055 ± 0.039	0.83	0.060 ± 0.062	0.53	0.54
2.0	Ramp	0.08	0.79	0.068 ± 0.083	0.80	0.071 ± 0.098	0.42	0.43
2.0	Exp	0.22	0.81	0.054 ± 0.065	0.85	0.055 ± 0.059	0.49	0.50
2.0	Sine	0.01	0.84	0.069 ± 0.084	0.91	0.080 ± 0.089	0.72	0.71
1.5	Step	0.24	0.82	0.052 ± 0.028	0.86	0.056 ± 0.049	0.52	0.52
1.5	Ramp	0.11	0.74	0.058 ± 0.053	0.77	0.060 ± 0.064	0.34	0.37
1.5	Exp	0.22	0.74	0.055 ± 0.048	0.74	0.056 ± 0.051	0.30	0.31
1.5	Sine	0.01	0.82	0.061 ± 0.065	0.89	0.075 ± 0.078	0.58	0.57
1.0	Step	0.22	0.80	0.050 ± 0.026	0.84	0.052 ± 0.029	0.52	0.52
1.0	Ramp	0.16	0.75	0.055 ± 0.037	0.75	0.054 ± 0.040	0.24	0.26
1.0	Exp	0.13	0.64	0.050 ± 0.027	0.64	0.050 ± 0.029	0.15	0.18
1.0	Sine	0.00	0.83	0.057 ± 0.060	0.91	0.069 ± 0.069	0.62	0.63
0.5	Step	0.24	0.83	0.050 ± 0.032	0.82	0.050 ± 0.029	0.40	0.41
0.5	Ramp	0.18	0.65	0.047 ± 0.021	0.66	0.048 ± 0.024	0.13	0.14
0.5	Exp	0.13	0.65	0.045 ± 0.021	0.65	0.046 ± 0.029	0.16	0.17
0.5	Sine	0.00	0.85	0.054 ± 0.052	0.91	0.061 ± 0.056	0.56	0.56
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = -0.3$								
None		0.00	0.93	0.032 ± 0.005	0.93	0.032 ± 0.005	0.95	0.86
2.0	Step	0.31	0.54	0.044 ± 0.019	0.60	0.047 ± 0.028	0.25	0.24
2.0	Ramp	0.16	0.51	0.044 ± 0.032	0.51	0.044 ± 0.035	0.15	0.14
2.0	Exp	0.16	0.49	0.041 ± 0.029	0.48	0.041 ± 0.028	0.15	0.13
2.0	Sine	0.00	0.83	0.059 ± 0.066	0.90	0.066 ± 0.067	0.56	0.50
1.5	Step	0.20	0.44	0.042 ± 0.020	0.52	0.046 ± 0.026	0.22	0.22
1.5	Ramp	0.18	0.45	0.039 ± 0.018	0.47	0.040 ± 0.023	0.14	0.12
1.5	Exp	0.17	0.39	0.037 ± 0.013	0.39	0.038 ± 0.015	0.05	0.05
1.5	Sine	0.00	0.82	0.053 ± 0.060	0.92	0.064 ± 0.064	0.56	0.52
1.0	Step	0.31	0.54	0.037 ± 0.015	0.56	0.038 ± 0.015	0.27	0.26
1.0	Ramp	0.19	0.40	0.036 ± 0.011	0.41	0.036 ± 0.013	0.08	0.07
1.0	Exp	0.15	0.39	0.035 ± 0.011	0.40	0.036 ± 0.013	0.06	0.05
1.0	Sine	0.00	0.82	0.042 ± 0.034	0.93	0.053 ± 0.044	0.53	0.49
0.5	Step	0.06	0.40	0.034 ± 0.007	0.40	0.034 ± 0.005	0.13	0.11
0.5	Ramp	0.07	0.32	0.033 ± 0.007	0.32	0.033 ± 0.007	0.05	0.04
0.5	Exp	0.13	0.43	0.035 ± 0.012	0.44	0.036 ± 0.018	0.09	0.07
0.5	Sine	0.00	0.90	0.046 ± 0.047	0.94	0.046 ± 0.040	0.66	0.63

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 31. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient $\Delta\mu_y(\sigma_y)$	Type	Fail	NOBM		OBM		Bias Test	
			Cov	Half Width	Cov	Half Width	$F(\hat{\sigma}_y^2)$	$\chi^2(\sigma_y^2)$
AR(2) Data with $\phi_1 = -1.0$ and $\phi_2 = -0.5$								
None		0.01	0.94	0.025 ± 0.006	0.94	0.025 ± 0.005	1.00	0.83
2.0	Step	0.52	0.38	0.044 ± 0.018	0.36	0.043 ± 0.017	0.24	0.09
2.0	Ramp	0.32	0.52	0.053 ± 0.044	0.45	0.048 ± 0.032	0.54	0.06
2.0	Exp	0.48	0.39	0.043 ± 0.036	0.37	0.041 ± 0.026	0.55	0.07
2.0	Sine	0.00	0.92	0.067 ± 0.068	0.94	0.071 ± 0.068	0.96	0.58
1.5	Step	0.30	0.25	0.044 ± 0.013	0.29	0.045 ± 0.015	0.29	0.08
1.5	Ramp	0.22	0.40	0.047 ± 0.030	0.43	0.048 ± 0.032	0.57	0.04
1.5	Exp	0.32	0.40	0.043 ± 0.031	0.39	0.042 ± 0.029	0.65	0.04
1.5	Sine	0.00	0.90	0.059 ± 0.057	0.97	0.065 ± 0.057	0.93	0.45
1.0	Step	0.36	0.27	0.039 ± 0.016	0.31	0.040 ± 0.015	0.51	0.08
1.0	Ramp	0.18	0.27	0.039 ± 0.017	0.35	0.043 ± 0.027	0.67	0.02
1.0	Exp	0.34	0.28	0.034 ± 0.013	0.32	0.036 ± 0.019	0.68	0.02
1.0	Sine	0.00	0.86	0.045 ± 0.043	0.96	0.053 ± 0.043	0.92	0.55
0.5	Step	0.28	0.32	0.031 ± 0.010	0.31	0.031 ± 0.010	0.95	0.09
0.5	Ramp	0.32	0.21	0.029 ± 0.005	0.21	0.029 ± 0.006	0.94	0.00
0.5	Exp	0.39	0.30	0.032 ± 0.014	0.33	0.035 ± 0.022	0.74	0.02
0.5	Sine	0.00	0.96	0.040 ± 0.035	0.97	0.040 ± 0.029	0.87	0.48
AR(2) Data with $\phi_1 = -1.5$ and $\phi_2 = -0.8$								
None		0.10	0.98	0.025 ± 0.007	0.98	0.025 ± 0.005	1.00	0.60
2.0	Step	0.32	0.36	0.074 ± 0.031	0.29	0.071 ± 0.034	0.13	0.04
2.0	Ramp	0.13	0.34	0.072 ± 0.066	0.46	0.077 ± 0.082	0.34	0.03
2.0	Exp	0.34	0.32	0.059 ± 0.024	0.48	0.065 ± 0.028	0.12	0.03
2.0	Sine	0.01	0.93	0.082 ± 0.066	0.95	0.084 ± 0.065	0.99	0.26
1.5	Step	0.30	0.31	0.066 ± 0.023	0.39	0.069 ± 0.026	0.11	0.03
1.5	Ramp	0.17	0.24	0.060 ± 0.043	0.37	0.069 ± 0.072	0.38	0.02
1.5	Exp	0.38	0.36	0.053 ± 0.028	0.37	0.054 ± 0.027	0.22	0.04
1.5	Sine	0.00	0.97	0.067 ± 0.047	0.98	0.068 ± 0.047	0.98	0.32
1.0	Step	0.59	0.40	0.052 ± 0.016	0.44	0.053 ± 0.018	0.10	0.02
1.0	Ramp	0.35	0.26	0.048 ± 0.021	0.27	0.048 ± 0.021	0.42	0.01
1.0	Exp	0.31	0.26	0.047 ± 0.021	0.25	0.046 ± 0.018	0.54	0.01
1.0	Sine	0.00	0.99	0.060 ± 0.042	0.99	0.061 ± 0.043	1.00	0.34
0.5	Step	0.15	0.25	0.045 ± 0.011	0.25	0.045 ± 0.011	0.63	0.02
0.5	Ramp	0.18	0.17	0.039 ± 0.009	0.19	0.040 ± 0.009	0.82	0.01
0.5	Exp	0.30	0.18	0.039 ± 0.014	0.18	0.040 ± 0.013	0.72	0.00
0.5	Sine	0.00	0.95	0.045 ± 0.029	0.99	0.049 ± 0.028	0.96	0.33

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

assumed to be known exactly and Schruben's χ^2_3 statistic in Equation (9) is tested. The assumption of known variance of the mean estimator variance is reasonable since Fishman's approximation is asymptotically unbiased and large samples are used. With the estimated variance $\hat{\sigma}^2_{\hat{\mu}_y}$ from Equation (56) and degrees of freedom d from Equation (57), Schruben's $F_{3,d}$ statistic, shown in Equation (10), is used.

The acceptance rate for Schruben's test for initial bias is very close to the nominal rate of 0.9 when the variance of the mean estimator, $\sigma^2_{\hat{\mu}_y}$, calculated with the true parameters, is used. The results are slightly less accurate when the estimated variance of the mean estimator $\hat{\sigma}^2_{\hat{\mu}_y}$ is used. However, the estimated truncation points generally result in a pass rate in excess of 50 percent for Schruben's initial bias test.

Schruben's initial bias identification test appears to be more dependent than the confidence interval techniques on the type of variation in the data. For the sequence without a transient, Schruben's technique worked well with the "true" variance estimate $\sigma^2_{\hat{\mu}_y}$ and the χ^2 test and also with the estimated mean estimator variance $\hat{\sigma}^2_{\hat{\mu}_y}$ and the F test. Since the estimated truncation point \hat{n}_0 usually retained some transient data in the sequence, Schruben's initial bias test appears to be extremely sensitive to even a small number of transient data values.

Tables 32, 33, 34, and 35 show the number and size of the batches of observations used for Nonoverlapping and Overlapping Batch Means confidence interval techniques. For the last two cases, $(\phi_1 = -1.0, \phi_2 = -0.5)$ and $(\phi_1 = -1.5, \phi_2 = -0.8)$, the average number of the batches for the sequence truncated at the estimated truncation point \hat{n}_0 significantly increases in comparison with the sequence truncated at the true point n_0 . Because of the increased variance in the batches' sample means, the confidence interval half widths actually increase. However, the batches which included transient values bias the mean estimate enough to cause a significant decrease in the coverage rate. These two cases correspond to the cases with significant degradation in actual coverage rates.

While the MSE appears relatively equal between the AR(1) and the AR(2) Kalman

Table 32. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model

Transient		Number of Batches		Size	
$\Delta\mu_y(\sigma_y)$	Type	$\hat{\mu}_M$	$\hat{\sigma}_M$	$\hat{\mu}_B$	$\hat{\sigma}_B$
AR(2) Data with $\phi_1 = 1.5$ and $\phi_2 = -0.8$					
None		66.84	34.34	19.31	11.56
2.0	Step	64.96	33.45	19.26	11.83
2.0	Ramp	65.59	32.02	18.87	11.05
2.0	Exp	64.65	33.97	19.21	11.29
2.0	Sine	65.30	32.44	18.91	11.62
1.5	Step	63.57	32.02	19.63	11.71
1.5	Ramp	67.13	33.89	18.49	10.46
1.5	Exp	63.55	33.15	19.87	12.37
1.5	Sine	66.30	35.97	18.97	11.41
1.0	Step	63.95	35.04	19.31	11.25
1.0	Ramp	67.94	34.36	18.75	11.52
1.0	Exp	64.77	33.43	19.23	11.57
1.0	Sine	65.44	34.47	19.28	11.83
0.5	Step	64.51	33.82	19.29	11.67
0.5	Ramp	75.16	39.26	17.50	9.71
0.5	Exp	79.02	41.00	16.98	10.63
0.5	Sine	83.39	40.70	16.39	9.64
AR(2) Data with $\phi_1 = 1.0$ and $\phi_2 = -0.5$					
None		168.98	68.09	7.35	4.88
2.0	Step	164.99	67.84	7.42	5.21
2.0	Ramp	174.30	70.96	7.17	5.31
2.0	Exp	162.46	70.27	7.88	6.67
2.0	Sine	167.60	70.92	7.51	5.35
1.5	Step	160.24	69.09	7.86	5.93
1.5	Ramp	177.78	73.96	7.47	7.09
1.5	Exp	175.47	73.76	7.38	6.13
1.5	Sine	189.83	77.34	7.22	7.09
1.0	Step	156.72	68.21	8.03	6.24
1.0	Ramp	183.77	77.42	7.42	6.56
1.0	Exp	197.14	79.22	6.91	5.24
1.0	Sine	214.87	81.92	6.69	6.52
0.5	Step	171.42	73.45	7.63	7.05
0.5	Ramp	208.13	85.34	7.54	10.63
0.5	Exp	207.53	84.77	7.47	9.51
0.5	Sine	218.39	88.33	6.86	4.32

Table 33. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model

Transient $\Delta\mu_y(\sigma_y)$ Type		Number of Batches $\hat{\mu}_M$ $\hat{\sigma}_M$		Size $\hat{\mu}_B$ $\hat{\sigma}_B$	
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = 0.3$					
None		40.87	16.95	29.60	14.52
2.0	Step	40.61	16.88	28.50	14.32
2.0	Ramp	39.15	17.28	31.23	16.30
2.0	Exp	40.43	17.19	29.25	15.47
2.0	Sine	39.04	17.67	31.80	17.56
1.5	Step	41.40	16.91	28.08	14.16
1.5	Ramp	37.78	17.94	33.99	18.24
1.5	Exp	37.51	17.42	33.19	19.58
1.5	Sine	34.36	15.61	36.60	17.99
1.0	Step	40.61	17.10	28.88	14.47
1.0	Ramp	35.49	17.20	37.85	23.65
1.0	Exp	31.89	17.35	45.92	31.55
1.0	Sine	35.82	14.55	36.56	19.36
0.5	Step	38.93	16.95	31.26	18.11
0.5	Ramp	30.94	18.98	55.14	41.21
0.5	Exp	31.09	18.80	55.57	39.66
0.5	Sine	38.58	16.31	37.46	20.78
AR(2) Data with $\phi_1 = 0.5$ and $\phi_2 = -0.3$					
None		475.92	81.83	2.30	1.50
2.0	Step	429.33	151.07	6.66	15.12
2.0	Ramp	472.95	138.71	8.22	26.71
2.0	Exp	468.05	115.58	4.67	16.39
2.0	Sine	432.80	180.61	8.73	19.15
1.5	Step	445.72	132.26	5.25	12.88
1.5	Ramp	493.25	120.43	5.30	19.44
1.5	Exp	485.70	133.74	5.91	20.92
1.5	Sine	457.39	194.57	8.64	18.48
1.0	Step	457.92	110.34	3.82	11.36
1.0	Ramp	501.31	132.86	5.23	18.95
1.0	Exp	539.92	108.01	3.18	10.96
1.0	Sine	489.74	202.53	8.41	18.72
0.5	Step	483.53	112.11	2.91	8.12
0.5	Ramp	557.20	129.24	3.17	9.91
0.5	Exp	568.61	130.48	4.21	15.54
0.5	Sine	549.51	191.41	6.36	15.39

Table 34. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model

Transient		Number of Batches		Size	
$\Delta\mu_y(\sigma_y)$	Type	$\hat{\mu}_M$	$\hat{\sigma}_M$	$\hat{\mu}_B$	$\hat{\sigma}_B$
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = 0.3$					
None		297.99	128.07	4.14	2.49
2.0	Step	229.83	157.99	15.29	22.46
2.0	Ramp	211.03	164.16	31.14	47.54
2.0	Exp	228.29	155.45	18.28	31.35
2.0	Sine	181.07	172.09	22.61	24.97
1.5	Step	232.60	159.65	15.00	22.09
1.5	Ramp	212.84	157.33	28.62	46.06
1.5	Exp	218.01	154.06	21.50	37.10
1.5	Sine	173.49	187.04	24.28	26.72
1.0	Step	222.75	153.79	15.25	22.34
1.0	Ramp	212.09	157.68	30.68	48.91
1.0	Exp	219.38	149.14	22.44	40.65
1.0	Sine	160.39	156.04	25.25	26.85
0.5	Step	240.67	151.38	15.90	31.57
0.5	Ramp	222.48	148.70	24.72	44.67
0.5	Exp	259.64	156.71	17.57	36.14
0.5	Sine	195.64	171.31	21.65	24.93
AR(2) Data with $\phi_1 = -0.5$ and $\phi_2 = -0.3$					
None		107.19	46.84	11.91	8.02
2.0	Step	212.61	206.70	19.16	23.41
2.0	Ramp	234.33	198.47	22.82	42.08
2.0	Exp	236.64	170.62	14.73	30.84
2.0	Sine	310.66	451.15	20.14	21.79
1.5	Step	290.06	241.69	14.81	22.40
1.5	Ramp	240.87	181.80	16.27	34.04
1.5	Exp	253.94	168.89	11.19	25.27
1.5	Sine	195.87	266.29	23.30	24.03
1.0	Step	194.39	104.11	9.78	15.44
1.0	Ramp	227.86	133.96	12.44	26.82
1.0	Exp	255.36	211.82	11.55	24.62
1.0	Sine	241.21	352.77	23.51	25.94
0.5	Step	219.75	98.57	7.55	10.20
0.5	Ramp	237.25	102.55	8.80	17.09
0.5	Exp	282.18	287.65	13.66	29.03
0.5	Sine	201.47	211.65	17.22	18.44

Table 35. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model

Transient		Number of Batches		Size	
$\Delta\mu_y(\sigma_y)$	Type	$\hat{\mu}_M$	$\hat{\sigma}_M$	$\hat{\mu}_B$	$\hat{\sigma}_B$
AR(2) Data with $\phi_1 = 1.5$ and $\phi_2 = -0.8$					
None		60.34	31.08	21.85	13.33
2.0	Step	317.37	216.50	11.14	16.50
2.0	Ramp	208.59	245.27	46.45	55.52
2.0	Exp	291.38	252.55	16.70	31.95
2.0	Sine	178.12	236.70	23.59	19.57
1.5	Step	355.32	239.43	12.56	19.84
1.5	Ramp	228.93	263.72	48.27	59.28
1.5	Exp	210.48	227.98	32.53	49.95
1.5	Sine	151.46	189.62	23.50	19.67
1.0	Step	336.94	255.74	12.70	21.06
1.0	Ramp	257.46	276.15	39.39	56.80
1.0	Exp	214.46	218.30	24.49	43.57
1.0	Sine	179.62	238.60	26.23	24.31
0.5	Step	132.29	43.17	12.97	19.14
0.5	Ramp	151.41	36.53	11.14	15.95
0.5	Exp	154.54	148.30	30.47	46.47
0.5	Sine	259.38	254.61	14.46	13.83
AR(2) Data with $\phi_1 = -1.5$ and $\phi_2 = -0.8$					
None		25.03	9.58	45.78	17.21
2.0	Step	172.44	116.79	21.56	25.43
2.0	Ramp	97.57	87.70	47.36	54.97
2.0	Exp	147.51	106.44	21.21	24.76
2.0	Sine	113.33	103.97	21.73	16.05
1.5	Step	165.98	116.30	20.67	24.63
1.5	Ramp	107.06	92.99	42.81	53.61
1.5	Exp	147.53	88.59	13.46	17.65
1.5	Sine	119.41	120.04	22.05	15.93
1.0	Step	133.48	77.89	13.85	15.78
1.0	Ramp	133.03	100.87	23.15	34.57
1.0	Exp	123.14	84.76	17.17	22.94
1.0	Sine	100.59	108.34	24.40	14.74
0.5	Step	115.29	45.20	13.82	15.69
0.5	Ramp	99.57	48.23	19.57	24.21
0.5	Exp	107.20	68.93	19.52	22.99
0.5	Sine	66.26	54.48	31.47	19.36

filter formulations, the truncation point selected by the AR(2) algorithm generally result in positive truncation-point errors. The AR(2) with measurement noise formulation almost always results in better confidence interval coverage rates and a significantly higher pass rate on Schruben's initial bias test. Since these tests are conducted with pure AR(2) data, it is not surprising that the AR(2) with measurement noise formulation performs better. The improvement may also be attributed to the increased flexibility of the AR(2) with measurement noise model. Because of the improved performance of the AR(2) with measurement noise formulation, only that model is applied in subsequent tests.

4.6.2 Autoregressive-Moving Average Test Cases. A similar Monte Carlo analysis is conducted on data sets generated from first-order autoregressive-moving average ARMA(1,1) models. The ARMA(1,1) model is selected because it is a simple time-series model from which data is generated easily, but the ARMA(1,1) data can not be fit precisely with an AR(1) or AR(2) formulation for the Kalman filter. Therefore, these data sets test the robustness of the approach. Since the assumed filter model does not fit the data, some adjustment for "lack of fit" is necessary. When actual discrete-event simulation output, which has an unknown analytical formulation, is modeled, the truncation-point selection algorithm needs to perform with some "lack of fit". The ARMA(1,1) model used to generate the data is

$$y_n - \mu_y = \phi_1(y_{n-1} - \mu_y) + w_d(t_n) - \theta_1 w_d(t_{n-1})$$

with variance of

$$\sigma_y^2 = \left(\frac{1 + \theta_1^2 - 2\phi_1\theta_1}{1 - \phi_1^2} \right) Q_d$$

Table 36 shows the parameters used to generate the data sequences $\{y_n\}$. These parameters are selected to test a wide range of the ARMA(1,1) stable region.

As seen in Table 37, for some combinations of parameters, the model parameters often failed to meet the model estimation criterion, Equation (58), or required a significant number of observations to estimate the model. This indicates that the AR(2) model embedded within the Kalman filter is unable to adjust to the correlation in ARMA(1,1) data.

Table 36. ARMA(1,1) Parameters for Generated Data

ϕ_1	θ_1	$\hat{\sigma}_y^2$	$\rho_y(1)$
-0.8	+0.3	4.36	-0.87
-0.8	-0.3	1.69	-0.62
-0.3	+0.8	2.12	-0.64
-0.3	-0.8	1.27	+0.33
+0.3	+0.8	1.27	-0.33
+0.3	-0.8	2.12	0.64
+0.8	+0.3	1.69	0.62
+0.8	-0.3	4.36	0.87

Table 37. Kalman Filter Model Estimation Summary

Data Parameters	Fail	Observations Used To Estimate				Cusum Alarms					
		Min	Max	Avg	St Dev	Min	Max	Avg	St Dev		
ARMA(1,1) Data with AR(2) with Measurement Noise Model											
$\phi_1 = -0.8 \quad \theta_1 = 0.3$	0.324	296	1499	1154.7	226.1	0	10	2.2	0.7		
$\phi_1 = -0.8 \quad \theta_1 = -0.3$	0.110	267	1499	904.9	390.6	0	20	3.8	2.3		
$\phi_1 = -0.3 \quad \theta_1 = 0.8$	0.344	470	1491	1230.1	124.2	0	10	2.1	0.6		
$\phi_1 = -0.3 \quad \theta_1 = -0.8$	0.015	256	1499	516.0	303.6	0	21	6.2	2.9		
$\phi_1 = 0.3 \quad \theta_1 = 0.8$	0.261	346	1499	1188.5	145.5	0	15	2.1	0.8		
$\phi_1 = 0.3 \quad \theta_1 = -0.8$	0.001	253	1414	335.0	105.6	2	23	7.2	3.1		
$\phi_1 = 0.8 \quad \theta_1 = 0.3$	0.001	252	1492	306.8	74.8	0	17	5.4	2.4		
$\phi_1 = 0.8 \quad \theta_1 = -0.3$	0.044	252	1494	385.3	205.3	0	21	5.8	2.8		

Note: Each row represents 16,000 runs: 1,000 for each transient type and magnitude.

Table 38 shows the estimated parameters for fitting an AR(2) model with measurement noise to ARMA(1,1) data. Based on the parameter estimates, the variance estimates in Table 39 are calculated. The variance of the measurement noise R should represent the "lack of fit". Since the ARMA(1,1) data is generated with dynamics noise variance Q_d of 1.0, the dynamics noise is slightly underestimated.

Tables 40 through 43 show the selected truncation-point errors, $(\hat{n}_o - n_o)$. The average truncation error is negative, meaning that generally the estimated truncation point \hat{n}_o retains transient data in the supposedly steady-state sequence. However, for the ARMA(1,1) case

Table 38. AR(2) with Measurement Noise Model Parameter Estimation Summary

Data Parameters	$\hat{\mu}_y$		$\hat{\phi}_1$		$\hat{\phi}_2$		\hat{k}	
	$\hat{\mu}_{\hat{\mu}_y}$	$\hat{\sigma}_{\hat{\mu}_y}$	$\hat{\mu}_{\hat{\phi}_1}$	$\hat{\sigma}_{\hat{\phi}_1}$	$\hat{\mu}_{\hat{\phi}_2}$	$\hat{\sigma}_{\hat{\phi}_2}$	$\hat{\mu}_{\hat{k}}$	$\hat{\sigma}_{\hat{k}}$
ARMA(1,1) Data with AR(2) with Measurement Noise Model								
$\phi_1 = -0.8 \quad \theta_1 = 0.3$	9.93	0.09	-1.19	0.14	-0.33	0.12	0.77	0.14
$\phi_1 = -0.8 \quad \theta_1 = -0.3$	9.96	0.07	-0.56	0.19	0.19	0.16	0.78	0.12
$\phi_1 = -0.3 \quad \theta_1 = 0.8$	9.91	0.09	-0.90	0.10	-0.33	0.07	0.74	0.19
$\phi_1 = -0.3 \quad \theta_1 = -0.8$	9.99	0.08	0.44	0.07	-0.24	0.06	0.90	0.00
$\phi_1 = 0.3 \quad \theta_1 = 0.8$	9.92	0.09	-0.38	0.19	-0.16	0.17	0.68	0.29
$\phi_1 = 0.3 \quad \theta_1 = -0.8$	10.00	0.14	0.91	0.06	-0.39	0.05	0.90	0.00
$\phi_1 = 0.8 \quad \theta_1 = 0.3$	10.00	0.20	0.67	0.13	0.08	0.10	0.73	0.12
$\phi_1 = 0.8 \quad \theta_1 = -0.3$	10.00	0.35	1.17	0.06	-0.35	0.06	0.90	0.00

Note: Each row represents the runs that did not fail to estimate a model from 16,000 runs.

Table 39. AR(2) with Measurement Noise Model Variance Estimation Summary

Data Parameters				\hat{Q}_d		\hat{R}		$\hat{\sigma}_y$		σ_y
				$\hat{\mu}_{\hat{Q}_d}$	$\hat{\sigma}_{\hat{Q}_d}$	$\hat{\mu}_{\hat{R}}$	$\hat{\sigma}_{\hat{R}}$	$\hat{\mu}_{\hat{\sigma}_y}$	$\hat{\sigma}_{\hat{\sigma}_y}$	
ARMA(1,1) Data with AR(2) with Measurement Noise Model										
$\phi_1 =$	-0.8	$\theta_1 =$	0.3	0.73	0.23	0.39	0.36	2.12	0.21	2.09
$\phi_1 =$	-0.8	$\theta_1 =$	-0.3	0.76	0.23	0.22	0.12	1.32	0.08	1.30
$\phi_1 =$	-0.3	$\theta_1 =$	0.8	0.98	0.26	0.52	0.53	1.58	0.10	1.53
$\phi_1 =$	-0.3	$\theta_1 =$	-0.8	0.96	0.08	0.11	0.02	1.13	0.05	1.13
$\phi_1 =$	0.3	$\theta_1 =$	0.8	0.89	0.50	0.52	0.68	1.23	0.16	1.13
$\phi_1 =$	0.3	$\theta_1 =$	-0.8	0.99	0.09	0.12	0.01	1.48	0.09	1.53
$\phi_1 =$	0.8	$\theta_1 =$	0.3	0.63	0.17	0.27	0.12	1.28	0.09	1.30
$\phi_1 =$	0.8	$\theta_1 =$	-0.3	0.78	0.07	0.10	0.01	2.00	0.72	2.09

Note: Each row represents the runs that did not fail to estimate a model from 16,000 runs.

Table 40. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
ARMA(1,1) Data with $\phi_1 = -0.8$ and $\theta_1 = 0.3$							
2.0	Step	0.000	-150	56	-57.5	44.5	5284.4
2.0	Ramp	0.000	-290	41	-154.7	75.8	29669.4
2.0	Exp	0.000	-248	53	-103.5	70.1	15613.1
2.0	Sine	0.015	-480	40	-193.3	142.1	57570.4
1.5	Step	0.000	-211	41	-83.7	60.7	10698.0
1.5	Ramp	0.000	-340	26	-174.0	86.2	37701.6
1.5	Exp	0.000	-314	44	-139.9	90.4	27748.6
1.5	Sine	0.042	-485	17	-256.3	155.3	89815.6
1.0	Step	0.000	-366	30	-133.4	97.5	27311.1
1.0	Ramp	0.000	-432	28	-218.6	103.1	58397.9
1.0	Exp	0.000	-392	10	-187.6	96.8	44547.1
1.0	Sine	0.088	-486	-4	-315.5	137.2	118340.8
0.5	Step	0.103	-471	33	-170.9	132.1	46639.9
0.5	Ramp	0.025	-486	-33	-282.3	100.5	89819.9
0.5	Exp	0.000	-464	-22	-293.2	88.9	93881.2
0.5	Sine	0.104	-486	-124	-366.3	102.3	144627.6
ARMA(1,1) Data with $\phi_1 = -0.8$ and $\theta_1 = -0.3$							
2.0	Step	0.007	-447	80	-57.8	82.1	10074.9
2.0	Ramp	0.000	-401	398	-137.3	96.0	28063.3
2.0	Exp	0.000	-371	410	-105.4	96.2	20377.5
2.0	Sine	0.027	-483	178	-212.2	153.6	68605.9
1.5	Step	0.046	-466	62	-68.6	100.4	14776.9
1.5	Ramp	0.000	-454	-7	-165.2	98.2	36931.9
1.5	Exp	0.000	-409	216	-162.1	92.0	34752.8
1.5	Sine	0.034	-483	25	-250.0	127.7	78812.6
1.0	Step	0.097	-467	56	-79.2	110.5	18482.4
1.0	Ramp	0.013	-476	494	-205.8	99.5	52251.7
1.0	Exp	0.000	-450	639	-218.1	86.0	54982.9
1.0	Sine	0.039	-483	447	-296.6	105.2	99032.2
0.5	Step	0.443	-461	288	-177.4	135.8	49910.7
0.5	Ramp	0.042	-477	372	-309.3	85.9	103019.6
0.5	Exp	0.000	-491	460	-297.9	78.4	94900.2
0.5	Sine	0.062	-484	489	-336.5	78.6	119409.4

Note: Negative Truncation points indicate transient observations are retained.

Table 41. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
ARMA(1,1) Data with $\phi_1 = -0.3$ and $\theta_1 = 0.8$							
2.0	Step	0.000	-139	-17	-69.1	39.2	6307.4
2.0	Ramp	0.000	-280	-86	-171.9	58.6	32987.2
2.0	Exp	0.000	-259	-48	-136.6	65.7	22989.9
2.0	Sine	0.005	-485	-34	-245.9	118.4	74463.5
1.5	Step	0.000	-194	-24	-104.4	55.0	13929.6
1.5	Ramp	0.000	-328	-100	-202.1	69.2	45622.7
1.5	Exp	0.000	-317	-58	-171.9	72.9	34875.7
1.5	Sine	0.009	-485	-64	-342.8	80.4	123989.7
1.0	Step	0.000	-324	-28	-153.1	92.2	31950.2
1.0	Ramp	0.000	-421	-76	-242.4	87.0	66317.2
1.0	Exp	0.000	-386	-123	-233.5	74.0	60019.6
1.0	Sine	0.008	-486	-187	-409.9	60.5	171717.8
0.5	Step	0.338	-465	-42	-142.4	81.4	26891.8
0.5	Ramp	0.001	-478	-189	-304.6	86.4	100269.6
0.5	Exp	0.000	-444	-227	-322.0	59.6	107244.4
0.5	Sine	0.009	-487	-313	-441.5	26.4	195630.4
ARMA(1,1) Data with $\phi_1 = -0.3$ and $\theta_1 = -0.8$							
2.0	Step	0.000	-403	593	3.5	71.9	5184.6
2.0	Ramp	0.000	-372	631	-94.8	87.2	16596.8
2.0	Exp	0.000	-321	677	-89.7	88.5	15874.2
2.0	Sine	0.002	-466	611	-155.0	106.3	35307.0
1.5	Step	0.005	-431	704	-1.9	89.5	8022.5
1.5	Ramp	0.000	-394	687	-127.0	98.6	25847.3
1.5	Exp	0.000	-380	689	-143.5	95.7	29761.7
1.5	Sine	0.001	-474	725	-204.3	118.9	55883.6
1.0	Step	0.035	-441	647	-42.6	103.7	12568.0
1.0	Ramp	0.004	-457	706	-182.4	113.4	46143.7
1.0	Exp	0.000	-427	715	-201.1	110.4	52645.8
1.0	Sine	0.011	-477	654	-256.2	116.9	79285.7
0.5	Step	0.524	-460	637	-148.7	181.0	54893.1
0.5	Ramp	0.177	-479	698	-295.1	151.4	110006.7
0.5	Exp	0.001	-488	647	-265.4	112.4	83057.0
0.5	Sine	0.009	-482	624	-310.2	116.0	109704.9

Note: Negative Truncation points indicate transient observations are retained.

Table 42. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
ARMA(1,1) Data with $\phi_1 = 0.3$ and $\theta_1 = 0.8$							
2.0	Step	0.000	-102	20	-53.3	28.8	3673.7
2.0	Ramp	0.000	-238	17	-136.2	54.0	21462.2
2.0	Exp	0.000	-233	8	-112.6	58.3	16081.4
2.0	Sine	0.010	-479	1	-205.1	117.6	55901.8
1.5	Step	0.000	-145	11	-71.7	40.3	6757.8
1.5	Ramp	0.000	-281	-7	-157.6	61.8	28647.6
1.5	Exp	0.000	-287	5	-149.7	67.2	26934.9
1.5	Sine	0.020	-488	0	-276.1	104.4	87138.5
1.0	Step	0.000	-227	5	-103.4	63.8	14776.6
1.0	Ramp	0.000	-360	-15	-195.2	73.8	43543.0
1.0	Exp	0.000	-345	-36	-198.2	65.8	43622.4
1.0	Sine	0.032	-488	-78	-333.7	82.2	118141.3
0.5	Step	0.175	-468	8	-149.6	103.8	33160.2
0.5	Ramp	0.046	-482	-108	-262.7	86.4	76462.5
0.5	Exp	0.000	-408	-73	-287.2	59.5	86012.8
0.5	Sine	0.109	-489	-183	-382.4	70.7	151217.3
ARMA(1,1) Data with $\phi_1 = 0.3$ and $\theta_1 = -0.8$							
2.0	Step	0.000	-41	737	41.1	124.8	17254.1
2.0	Ramp	0.000	-165	717	-26.6	133.8	18601.0
2.0	Exp	0.000	-135	726	22.8	133.2	18272.7
2.0	Sine	0.000	-215	702	-15.0	146.5	21682.0
1.5	Step	0.000	-81	732	31.7	106.6	12369.2
1.5	Ramp	0.000	-237	715	-40.4	153.8	25294.4
1.5	Exp	0.000	-224	732	-34.3	142.0	21327.7
1.5	Sine	0.000	-265	719	-82.8	157.2	31579.0
1.0	Step	0.000	-281	735	30.7	124.2	16360.0
1.0	Ramp	0.000	-289	738	-77.4	164.0	32889.7
1.0	Exp	0.000	-285	726	-103.2	164.5	37723.1
1.0	Sine	0.000	-324	737	-147.5	190.0	57842.3
0.5	Step	0.107	-448	740	-43.7	183.5	35583.3
0.5	Ramp	0.006	-457	738	-188.5	200.5	75740.0
0.5	Exp	0.000	-348	729	-180.8	195.7	70980.5
0.5	Sine	0.000	-445	734	-226.8	200.8	91777.4

Note: Negative Truncation points indicate transient observations are retained.

Table 43. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
ARMA(1,1) Data with $\phi_1 = 0.8$ and $\theta_1 = 0.3$							
2.0	Step	0.000	-2	726	23.4	89.3	8528.1
2.0	Ramp	0.000	-100	711	-13.7	90.9	8453.2
2.0	Exp	0.000	-51	701	19.9	86.6	7895.0
2.0	Sine	0.000	-82	734	9.7	96.3	9374.1
1.5	Step	0.000	-7	709	19.6	76.5	6236.7
1.5	Ramp	0.000	-140	741	-24.7	92.7	9204.8
1.5	Exp	0.000	-139	732	0.0	102.4	10480.5
1.5	Sine	0.000	-195	708	-33.5	121.9	15978.2
1.0	Step	0.000	-4	718	21.9	92.3	8989.9
1.0	Ramp	0.000	-167	721	-42.9	118.7	15937.5
1.0	Exp	0.000	-296	740	-67.5	119.3	18799.5
1.0	Sine	0.001	-256	735	-110.5	132.5	29777.7
0.5	Step	0.000	-332	674	11.8	84.2	7227.8
0.5	Ramp	0.000	-350	722	-120.4	125.2	30183.2
0.5	Exp	0.000	-290	738	-153.4	144.5	44410.2
0.5	Sine	0.000	-326	734	-197.6	158.1	64036.8
ARMA(1,1) Data with $\phi_1 = 0.8$ and $\theta_1 = -0.3$							
2.0	Step	0.000	-224	742	56.6	146.6	24688.3
2.0	Ramp	0.000	-390	738	27.9	171.3	30138.0
2.0	Exp	0.000	-298	738	57.0	155.9	27547.7
2.0	Sine	0.001	-441	742	40.5	153.9	25323.9
1.5	Step	0.001	-73	740	52.9	145.8	24055.8
1.5	Ramp	0.000	-211	731	13.8	168.3	28500.9
1.5	Exp	0.000	-243	741	47.9	153.7	25925.7
1.5	Sine	0.003	-473	744	31.2	180.0	33367.4
1.0	Step	0.006	-276	733	54.4	152.4	26185.5
1.0	Ramp	0.000	-346	739	-2.8	182.6	33343.7
1.0	Exp	0.000	-295	728	-0.6	169.3	28657.3
1.0	Sine	0.003	-478	740	-42.0	205.6	44038.4
0.5	Step	0.005	-346	741	52.7	168.4	31147.2
0.5	Ramp	0.000	-405	725	-72.9	200.8	45636.2
0.5	Exp	0.000	-371	742	-90.9	216.5	55140.1
0.5	Sine	0.002	-471	748	-136.4	241.4	76865.1

Note: Negative Truncation points indicate transient observations are retained.

with $\phi_1 = 0.8$ and $\theta_1 = -0.3$, the algorithm selected good truncation points \hat{n}_0 since the errors are on the average are about 50 observations after the completion of the induced transient.

For the ARMA(1,1) data, the coverage rate is less for the sequence truncated at the estimated truncation point \hat{n}_0 compared to the sequence truncation at the true truncation point n_0 . As before, coverage decreases as the shift in the mean decreases with the exception of the exponentially decaying sinusoidal transient. For that transient, low transient values and high transient values may cancel sufficiently to prevent a significant reduction in coverage rate. The decaying exponential sinusoidal transient also has a very high rate of passing Schruben's test for initial bias. The mixture of low and high transient values are apparently more difficult for that statistical test to detect.

The extent of the coverage decrease for the sequence truncated at the estimated point appears to depend on the correlation structure in the generated data. For the data sets with a positive theoretical autocorrelation at lag one (see Table 36), the decrease in coverage is small. For the same cases, the truncated sequences pass Schruben's test for initial bias at fairly high rates. Actual discrete-event simulation output sequences generally have high positive correlation, so these results are promising.

Tables 48 through 51 present the number and size of the batches used in the confidence interval construction techniques. For each case, these numbers for 1000 sequences truncated at the true truncation point n_0 are also shown. In general, the sequences truncated at the estimated point has about the same number of batches and batch sizes. For the four cases with positive autocorrelation at lag one, the case with $\phi_1 = -0.3$ and $\theta_1 = 0.8$ has the largest decrease in coverage. For this case, the number of batches increases dramatically from the sequences without a transient. The increase in the number of batches causes tight confidence intervals about a slightly biased mean estimate.

4.6.3 Analysis. The Monte Carlo results indicate that the technique works well for sequences which display a high positive autocorrelation and significant transients. The al-

Table 44. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient $\Delta\mu_y(\sigma_y)$ Type		Fail	NOBM Cov Half Width		OBM Cov Half Width		Bias Test $F(\hat{\sigma}_y^2)$
ARMA(1,1) Data with $\phi_1 = -0.8$ and $\theta_1 = 0.3$							
None		0.02	0.94	0.027 ± 0.009	0.95	0.028 ± 0.008	1.00
2.0	Step	0.65	0.65	0.052 ± 0.051	0.75	0.062 ± 0.082	0.24
2.0	Ramp	0.19	0.41	0.089 ± 0.107	0.52	0.101 ± 0.141	0.49
2.0	Exp	0.51	0.39	0.041 ± 0.012	0.57	0.049 ± 0.032	0.31
2.0	Sine	0.01	0.82	0.091 ± 0.105	0.90	0.098 ± 0.099	0.89
1.5	Step	0.68	0.78	0.050 ± 0.035	0.70	0.048 ± 0.039	0.23
1.5	Ramp	0.22	0.33	0.059 ± 0.066	0.46	0.067 ± 0.091	0.57
1.5	Exp	0.43	0.38	0.041 ± 0.038	0.38	0.041 ± 0.038	0.49
1.5	Sine	0.00	0.90	0.090 ± 0.096	0.96	0.096 ± 0.091	0.86
1.0	Step	0.61	0.44	0.042 ± 0.016	0.63	0.050 ± 0.034	0.26
1.0	Ramp	0.34	0.30	0.040 ± 0.026	0.36	0.043 ± 0.042	0.60
1.0	Exp	0.27	0.31	0.037 ± 0.016	0.33	0.038 ± 0.023	0.71
1.0	Sine	0.00	0.76	0.065 ± 0.076	0.98	0.085 ± 0.071	0.86
0.5	Step	0.19	0.28	0.037 ± 0.006	0.28	0.038 ± 0.006	0.67
0.5	Ramp	0.22	0.22	0.035 ± 0.005	0.23	0.035 ± 0.006	0.77
0.5	Exp	0.30	0.27	0.034 ± 0.007	0.27	0.034 ± 0.010	0.73
0.5	Sine	0.00	0.80	0.053 ± 0.059	0.97	0.067 ± 0.055	0.87
ARMA(1,1) Data with $\phi_1 = -0.8$ and $\theta_1 = -0.3$							
None		0.00	0.92	0.041 ± 0.003	0.93	0.041 ± 0.003	0.93
2.0	Step	0.33	0.83	0.048 ± 0.020	0.85	0.049 ± 0.027	0.42
2.0	Ramp	0.15	0.70	0.054 ± 0.046	0.73	0.057 ± 0.062	0.24
2.0	Exp	0.20	0.68	0.050 ± 0.045	0.69	0.051 ± 0.052	0.26
2.0	Sine	0.00	0.86	0.066 ± 0.073	0.91	0.075 ± 0.077	0.58
1.5	Step	0.30	0.82	0.047 ± 0.035	0.86	0.050 ± 0.035	0.42
1.5	Ramp	0.19	0.67	0.045 ± 0.026	0.69	0.046 ± 0.032	0.21
1.5	Exp	0.17	0.54	0.043 ± 0.020	0.54	0.043 ± 0.022	0.10
1.5	Sine	0.00	0.85	0.055 ± 0.055	0.91	0.065 ± 0.061	0.62
1.0	Step	0.28	0.77	0.044 ± 0.027	0.78	0.045 ± 0.024	0.42
1.0	Ramp	0.17	0.57	0.041 ± 0.013	0.58	0.042 ± 0.018	0.16
1.0	Exp	0.15	0.51	0.039 ± 0.011	0.52	0.040 ± 0.016	0.11
1.0	Sine	0.00	0.87	0.051 ± 0.049	0.93	0.057 ± 0.051	0.52
0.5	Step	0.04	0.49	0.040 ± 0.008	0.49	0.040 ± 0.008	0.17
0.5	Ramp	0.04	0.37	0.037 ± 0.006	0.37	0.037 ± 0.006	0.05
0.5	Exp	0.15	0.58	0.038 ± 0.011	0.58	0.039 ± 0.016	0.12
0.5	Sine	0.00	0.87	0.045 ± 0.035	0.91	0.047 ± 0.033	0.75

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 45. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient $\Delta\mu_y(\sigma_y)$ Type		Fail	NOBM Cov Half Width		OBM Cov Half Width		Bias Test $F(\hat{\sigma}_y^2)$
ARMA(1,1) Data with $\phi_1 = -0.3$ and $\theta_1 = 0.8$							
None		0.12	0.98	0.013 ± 0.004	0.98	0.014 ± 0.003	0.99
2.0	Step	0.79	0.11	0.050 ± 0.030	0.16	0.058 ± 0.046	0.00
2.0	Ramp	0.24	0.02	0.077 ± 0.091	0.22	0.098 ± 0.154	0.00
2.0	Exp	0.33	0.00	0.044 ± 0.010	0.07	0.046 ± 0.011	0.00
2.0	Sine	0.02	0.51	0.083 ± 0.123	0.93	0.109 ± 0.097	0.73
1.5	Step	0.85	0.24	0.057 ± 0.016	0.09	0.054 ± 0.011	0.00
1.5	Ramp	0.47	0.04	0.062 ± 0.059	0.20	0.067 ± 0.075	0.00
1.5	Exp	0.75	0.01	0.040 ± 0.021	0.01	0.041 ± 0.025	0.00
1.5	Sine	0.00	0.72	0.065 ± 0.090	0.96	0.092 ± 0.092	0.68
1.0	Step	0.34	0.02	0.048 ± 0.008	0.06	0.049 ± 0.012	0.00
1.0	Ramp	0.45	0.55	0.065 ± 0.041	0.40	0.053 ± 0.028	0.00
1.0	Exp	0.57	0.19	0.043 ± 0.029	0.15	0.040 ± 0.024	0.00
1.0	Sine	0.00	0.58	0.087 ± 0.116	0.99	0.125 ± 0.079	0.72
0.5	Step	0.69	0.02	0.036 ± 0.005	0.08	0.037 ± 0.007	0.00
0.5	Ramp	0.35	0.01	0.035 ± 0.012	0.25	0.043 ± 0.029	0.00
0.5	Exp	0.53	0.00	0.035 ± 0.015	0.02	0.036 ± 0.015	0.00
0.5	Sine	0.00	0.64	0.073 ± 0.080	0.99	0.103 ± 0.049	0.68
ARMA(1,1) Data with $\phi_1 = -0.3$ and $\theta_1 = -0.8$							
None		0.00	0.88	0.070 ± 0.007	0.88	0.069 ± 0.006	0.85
2.0	Step	0.06	0.85	0.072 ± 0.020	0.85	0.072 ± 0.023	0.76
2.0	Ramp	0.02	0.76	0.071 ± 0.025	0.76	0.071 ± 0.027	0.39
2.0	Exp	0.04	0.73	0.075 ± 0.037	0.72	0.074 ± 0.033	0.26
2.0	Sine	0.00	0.85	0.074 ± 0.035	0.86	0.076 ± 0.038	0.66
1.5	Step	0.06	0.87	0.072 ± 0.027	0.87	0.072 ± 0.025	0.73
1.5	Ramp	0.03	0.72	0.072 ± 0.026	0.72	0.072 ± 0.027	0.27
1.5	Exp	0.03	0.64	0.072 ± 0.028	0.63	0.071 ± 0.025	0.21
1.5	Sine	0.00	0.86	0.071 ± 0.028	0.86	0.073 ± 0.035	0.62
1.0	Step	0.09	0.82	0.075 ± 0.037	0.81	0.074 ± 0.033	0.54
1.0	Ramp	0.06	0.65	0.070 ± 0.023	0.64	0.070 ± 0.024	0.21
1.0	Exp	0.03	0.67	0.068 ± 0.020	0.67	0.068 ± 0.020	0.22
1.0	Sine	0.00	0.85	0.068 ± 0.025	0.86	0.070 ± 0.029	0.68
0.5	Step	0.03	0.64	0.070 ± 0.028	0.63	0.070 ± 0.025	0.29
0.5	Ramp	0.05	0.56	0.065 ± 0.019	0.56	0.065 ± 0.018	0.14
0.5	Exp	0.03	0.73	0.065 ± 0.015	0.73	0.064 ± 0.015	0.28
0.5	Sine	0.00	0.85	0.065 ± 0.020	0.85	0.066 ± 0.020	0.73

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 46. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient $\Delta\mu_y(\sigma_y)$	Type	Fail	NOBM		OBM		Bias Test $F(\hat{\sigma}_y^2)$
			Cov	Half Width	Cov	Half Width	
ARMA(1,1) Data with $\phi_1 = 0.3$ and $\theta_1 = 0.8$							
None		0.03	0.97	0.020 ± 0.005	0.98	0.020 ± 0.004	1.00
2.0	Step	0.51	0.22	0.057 ± 0.026	0.26	0.059 ± 0.030	0.03
2.0	Ramp	0.17	0.22	0.072 ± 0.109	0.23	0.067 ± 0.088	0.03
2.0	Exp	0.18	0.11	0.046 ± 0.014	0.13	0.047 ± 0.015	0.02
2.0	Sine	0.01	0.57	0.064 ± 0.116	0.84	0.100 ± 0.143	0.65
1.5	Step	0.43	0.13	0.057 ± 0.023	0.17	0.059 ± 0.027	0.02
1.5	Ramp	0.16	0.12	0.056 ± 0.061	0.16	0.059 ± 0.078	0.03
1.5	Exp	0.37	0.12	0.045 ± 0.034	0.14	0.047 ± 0.037	0.03
1.5	Sine	0.01	0.61	0.066 ± 0.119	0.85	0.098 ± 0.128	0.61
1.0	Step	0.46	0.19	0.055 ± 0.017	0.20	0.055 ± 0.018	0.02
1.0	Ramp	0.29	0.14	0.050 ± 0.043	0.19	0.053 ± 0.055	0.02
1.0	Exp	0.26	0.22	0.047 ± 0.033	0.18	0.046 ± 0.037	0.01
1.0	Sine	0.00	0.69	0.072 ± 0.127	0.84	0.090 ± 0.125	0.47
0.5	Step	0.39	0.13	0.040 ± 0.008	0.18	0.042 ± 0.010	0.02
0.5	Ramp	0.33	0.19	0.045 ± 0.023	0.30	0.048 ± 0.031	0.00
0.5	Exp	0.20	0.06	0.036 ± 0.016	0.13	0.042 ± 0.021	0.01
0.5	Sine	0.00	0.83	0.079 ± 0.129	0.91	0.093 ± 0.113	0.46
ARMA(1,1) Data with $\phi_1 = 0.3$ and $\theta_1 = -0.8$							
None		0.00	0.88	0.127 ± 0.020	0.88	0.127 ± 0.018	0.79
2.0	Step	0.00	0.86	0.130 ± 0.027	0.86	0.130 ± 0.025	0.84
2.0	Ramp	0.01	0.83	0.130 ± 0.033	0.82	0.129 ± 0.030	0.56
2.0	Exp	0.01	0.85	0.132 ± 0.033	0.85	0.131 ± 0.032	0.64
2.0	Sine	0.01	0.83	0.133 ± 0.031	0.83	0.133 ± 0.031	0.76
1.5	Step	0.00	0.87	0.129 ± 0.026	0.86	0.128 ± 0.024	0.84
1.5	Ramp	0.01	0.80	0.132 ± 0.042	0.79	0.130 ± 0.034	0.49
1.5	Exp	0.03	0.77	0.136 ± 0.051	0.77	0.134 ± 0.042	0.40
1.5	Sine	0.00	0.83	0.132 ± 0.033	0.84	0.131 ± 0.032	0.70
1.0	Step	0.01	0.86	0.130 ± 0.028	0.86	0.129 ± 0.027	0.81
1.0	Ramp	0.01	0.77	0.133 ± 0.047	0.77	0.132 ± 0.043	0.36
1.0	Exp	0.01	0.73	0.133 ± 0.045	0.73	0.132 ± 0.043	0.30
1.0	Sine	0.00	0.85	0.129 ± 0.031	0.85	0.128 ± 0.031	0.62
0.5	Step	0.08	0.76	0.138 ± 0.055	0.76	0.136 ± 0.050	0.42
0.5	Ramp	0.08	0.68	0.133 ± 0.053	0.68	0.131 ± 0.050	0.25
0.5	Exp	0.02	0.75	0.125 ± 0.040	0.75	0.124 ± 0.035	0.29
0.5	Sine	0.00	0.84	0.122 ± 0.031	0.84	0.121 ± 0.031	0.76

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 47. AR(2) with Measurement Noise Model Truncation Point \hat{n}_o Evaluations

Transient $\Delta\mu_y(\sigma_y)$	Type	Fail	NOBM		OBM		Bias Test $F(\hat{\sigma}_y^2)$
			Cov	Half Width	Cov	Half Width	
ARMA(1,1) Data with $\phi_1 = 0.8$ and $\theta_1 = 0.3$							
None		0.01	0.86	0.169 ± 0.046	0.85	0.164 ± 0.039	0.90
2.0	Step	0.01	0.85	0.170 ± 0.047	0.85	0.166 ± 0.040	0.90
2.0	Ramp	0.02	0.84	0.168 ± 0.047	0.83	0.165 ± 0.042	0.81
2.0	Exp	0.01	0.85	0.169 ± 0.048	0.85	0.165 ± 0.042	0.91
2.0	Sine	0.01	0.83	0.172 ± 0.054	0.83	0.170 ± 0.053	0.89
1.5	Step	0.01	0.86	0.168 ± 0.045	0.84	0.164 ± 0.040	0.93
1.5	Ramp	0.02	0.82	0.170 ± 0.051	0.81	0.168 ± 0.045	0.77
1.5	Exp	0.04	0.81	0.175 ± 0.056	0.80	0.172 ± 0.053	0.71
1.5	Sine	0.03	0.82	0.182 ± 0.055	0.83	0.183 ± 0.056	0.81
1.0	Step	0.01	0.84	0.169 ± 0.049	0.83	0.165 ± 0.042	0.93
1.0	Ramp	0.05	0.81	0.176 ± 0.064	0.80	0.172 ± 0.053	0.62
1.0	Exp	0.10	0.78	0.193 ± 0.098	0.77	0.188 ± 0.078	0.42
1.0	Sine	0.02	0.86	0.181 ± 0.052	0.86	0.181 ± 0.051	0.76
0.5	Step	0.03	0.84	0.172 ± 0.070	0.83	0.169 ± 0.062	0.86
0.5	Ramp	0.07	0.74	0.199 ± 0.101	0.75	0.194 ± 0.095	0.34
0.5	Exp	0.04	0.75	0.186 ± 0.080	0.74	0.183 ± 0.073	0.35
0.5	Sine	0.01	0.85	0.170 ± 0.049	0.85	0.169 ± 0.046	0.67
ARMA(1,1) Data with $\phi_1 = 0.8$ and $\theta_1 = -0.3$							
None		0.02	0.87	0.315 ± 0.087	0.86	0.306 ± 0.075	0.83
2.0	Step	0.02	0.82	0.319 ± 0.101	0.82	0.311 ± 0.097	0.86
2.0	Ramp	0.02	0.82	0.315 ± 0.115	0.82	0.311 ± 0.113	0.75
2.0	Exp	0.02	0.85	0.315 ± 0.097	0.84	0.307 ± 0.087	0.86
2.0	Sine	0.01	0.84	0.334 ± 0.229	0.83	0.329 ± 0.236	0.82
1.5	Step	0.02	0.84	0.314 ± 0.094	0.83	0.308 ± 0.096	0.86
1.5	Ramp	0.01	0.80	0.324 ± 0.147	0.80	0.318 ± 0.128	0.71
1.5	Exp	0.03	0.82	0.322 ± 0.156	0.81	0.313 ± 0.135	0.81
1.5	Sine	0.02	0.82	0.338 ± 0.198	0.81	0.336 ± 0.213	0.84
1.0	Step	0.02	0.84	0.318 ± 0.105	0.83	0.310 ± 0.096	0.85
1.0	Ramp	0.04	0.81	0.324 ± 0.125	0.80	0.319 ± 0.119	0.60
1.0	Exp	0.06	0.80	0.339 ± 0.156	0.80	0.333 ± 0.135	0.49
1.0	Sine	0.02	0.83	0.342 ± 0.172	0.82	0.344 ± 0.197	0.72
0.5	Step	0.04	0.82	0.325 ± 0.143	0.82	0.317 ± 0.128	0.78
0.5	Ramp	0.05	0.77	0.343 ± 0.158	0.77	0.337 ± 0.147	0.38
0.5	Exp	0.04	0.75	0.341 ± 0.148	0.74	0.335 ± 0.141	0.36
0.5	Sine	0.01	0.83	0.315 ± 0.108	0.83	0.317 ± 0.126	0.69

Note: Since the nominal rates are 0.9, the coverage and bias test accuracies are $\approx \pm 0.016$ for 1000 runs.

Table 48. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model

Transient $\Delta\mu_y(\sigma_y)$ Type		Number of Batches $\hat{\mu}_M$ $\hat{\sigma}_M$		Size $\hat{\mu}_B$ $\hat{\sigma}_B$	
ARMA(1,1) Data with $\phi_1 = -0.8$ and $\theta_1 = 0.3$					
None		90.69	121.40	23.92	16.74
2.0	Step	96.60	145.20	32.11	24.15
2.0	Ramp	238.32	253.60	58.63	62.84
2.0	Exp	307.29	237.96	21.75	27.34
2.0	Sine	174.48	214.16	26.73	20.71
1.5	Step	137.29	192.70	34.69	26.71
1.5	Ramp	322.00	255.01	38.84	57.14
1.5	Exp	410.30	206.82	9.26	22.45
1.5	Sine	135.80	185.58	27.30	18.64
1.0	Step	274.15	239.78	24.47	27.87
1.0	Ramp	385.35	236.77	17.30	38.13
1.0	Exp	398.26	213.89	8.32	21.59
1.0	Sine	111.08	166.99	35.51	25.97
0.5	Step	478.40	186.38	4.38	6.58
0.5	Ramp	433.29	227.19	6.32	14.72
0.5	Exp	389.66	244.51	7.57	13.80
0.5	Sine	143.93	206.44	32.13	24.42
ARMA(1,1) Data with $\phi_1 = -0.8$ and $\theta_1 = -0.3$					
None		445.35	117.92	2.76	2.45
2.0	Step	344.62	206.18	14.16	22.41
2.0	Ramp	406.93	218.30	27.53	51.60
2.0	Exp	433.84	184.01	16.68	40.21
2.0	Sine	304.61	250.19	20.06	24.34
1.5	Step	356.33	205.22	15.04	25.89
1.5	Ramp	459.19	190.77	17.07	41.27
1.5	Exp	490.26	163.07	10.64	31.62
1.5	Sine	335.32	261.54	20.82	27.09
1.0	Step	437.35	153.90	8.25	22.77
1.0	Ramp	509.72	170.01	10.79	32.45
1.0	Exp	545.61	134.65	5.01	18.48
1.0	Sine	389.81	273.28	18.36	26.69
0.5	Step	537.70	140.19	3.62	12.71
0.5	Ramp	612.63	122.00	2.95	9.62
0.5	Exp	553.72	182.83	9.57	29.77
0.5	Sine	459.79	269.77	12.75	19.98

Table 49. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model

Transient $\Delta\mu_y(\sigma_y)$ Type		Number of Batches $\hat{\mu}_M$ $\hat{\sigma}_M$		Size $\hat{\mu}_B$ $\hat{\sigma}_B$	
ARMA(1,1) Data with $\phi_1 = -0.3$ and $\theta_1 = 0.8$					
None		24.50	11.57	48.28	18.11
2.0	Step	366.52	212.91	12.89	20.88
2.0	Ramp	315.95	229.52	38.00	59.42
2.0	Exp	358.33	183.50	8.70	17.56
2.0	Sine	68.30	183.98	43.91	23.37
1.5	Step	353.73	210.38	15.26	24.89
1.5	Ramp	310.38	238.02	35.65	54.95
1.5	Exp	314.34	169.43	6.14	14.60
1.5	Sine	386.59	381.52	25.54	29.87
1.0	Step	422.17	185.46	6.03	13.71
1.0	Ramp	138.28	197.03	76.45	61.65
1.0	Exp	194.43	195.40	33.19	51.17
1.0	Sine	74.44	165.40	46.06	22.38
0.5	Step	237.27	91.12	8.80	14.75
0.5	Ramp	113.16	89.64	55.10	63.32
0.5	Exp	324.57	253.90	13.95	32.21
0.5	Sine	107.26	209.44	43.03	22.45
ARMA(1,1) Data with $\phi_1 = -0.3$ and $\theta_1 = -0.8$					
None		392.19	140.37	3.23	2.52
2.0	Step	367.54	156.43	5.53	11.69
2.0	Ramp	360.92	185.44	9.15	23.06
2.0	Exp	332.24	189.78	12.80	29.37
2.0	Sine	308.30	209.98	11.59	17.91
1.5	Step	360.68	158.45	5.54	12.74
1.5	Ramp	345.85	193.91	12.43	29.67
1.5	Exp	341.26	199.93	12.69	29.50
1.5	Sine	325.56	217.58	10.78	15.93
1.0	Step	343.03	170.85	9.70	25.60
1.0	Ramp	358.47	202.22	11.41	27.81
1.0	Exp	374.15	203.48	9.36	22.79
1.0	Sine	343.66	223.21	9.92	14.76
0.5	Step	384.52	190.47	9.06	24.39
0.5	Ramp	397.15	214.06	8.77	21.50
0.5	Exp	412.07	206.51	6.83	15.81
0.5	Sine	385.97	227.19	8.28	13.56

Table 50. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model

Transient		Number of Batches		Size	
$\Delta\mu_y(\sigma_y)$	Type	$\hat{\mu}_M$	$\hat{\sigma}_M$	$\hat{\mu}_B$	$\hat{\sigma}_B$
ARMA(1,1) Data with $\phi_1 = 0.3$ and $\theta_1 = 0.8$					
None		37.77	19.16	33.92	17.63
2.0	Step	666.50	376.69	6.16	12.44
2.0	Ramp	434.07	406.44	24.75	45.19
2.0	Exp	429.11	335.65	8.37	16.26
2.0	Sine	238.47	314.80	31.83	32.00
1.5	Step	695.71	399.02	6.59	14.33
1.5	Ramp	442.89	388.28	16.91	37.54
1.5	Exp	411.77	335.44	10.43	23.49
1.5	Sine	426.70	541.72	27.98	30.15
1.0	Step	591.16	453.60	12.54	22.39
1.0	Ramp	303.04	261.93	25.03	44.26
1.0	Exp	314.71	373.03	29.09	48.94
1.0	Sine	514.44	490.38	20.13	26.01
0.5	Step	242.58	128.39	11.71	19.31
0.5	Ramp	147.25	178.25	59.17	63.17
0.5	Exp	314.87	372.24	35.96	55.82
0.5	Sine	363.92	434.28	24.09	25.39
ARMA(1,1) Data with $\phi_1 = 0.3$ and $\theta_1 = -0.8$					
None		142.83	63.41	8.87	5.90
2.0	Step	141.27	63.18	8.61	6.30
2.0	Ramp	136.71	69.59	11.26	11.66
2.0	Exp	136.97	66.65	10.11	9.85
2.0	Sine	125.21	67.22	12.43	12.07
1.5	Step	146.11	63.60	8.27	5.47
1.5	Ramp	130.87	66.88	12.86	16.96
1.5	Exp	125.44	70.52	15.14	21.36
1.5	Sine	121.15	67.49	13.81	13.33
1.0	Step	142.13	64.53	8.83	7.02
1.0	Ramp	129.61	73.42	16.36	24.96
1.0	Exp	124.21	71.78	18.04	26.97
1.0	Sine	120.67	67.65	14.55	14.03
0.5	Step	127.29	71.98	16.78	26.35
0.5	Ramp	125.95	75.69	21.63	33.44
0.5	Exp	139.41	73.22	14.09	19.27
0.5	Sine	139.16	72.51	12.76	11.73

Table 51. Fishman's Number and Size of Batches for AR(2) with Measurement Noise Model

Transient		Number of Batches		Size	
$\Delta\mu_y(\sigma_y)$	Type	$\hat{\mu}_M$	$\hat{\sigma}_M$	$\hat{\mu}_B$	$\hat{\sigma}_B$
ARMA(1,1) Data with $\phi_1 = -0.8$ and $\theta_1 = 0.3$					
None		50.89	18.10	23.39	12.54
2.0	Step	50.68	18.47	22.82	12.09
2.0	Ramp	49.75	19.07	24.63	13.52
2.0	Exp	50.69	18.76	23.43	13.58
2.0	Sine	48.56	20.69	26.07	16.34
1.5	Step	51.58	18.44	22.66	12.56
1.5	Ramp	48.55	20.16	26.26	15.30
1.5	Exp	47.21	19.70	26.53	15.83
1.5	Sine	41.01	20.72	32.92	18.79
1.0	Step	50.68	19.35	23.27	12.99
1.0	Ramp	45.01	20.85	29.87	18.18
1.0	Exp	39.93	21.57	37.75	26.96
1.0	Sine	42.28	18.77	32.54	18.32
0.5	Step	49.88	18.77	24.17	14.87
0.5	Ramp	37.63	23.93	50.21	40.67
0.5	Exp	39.09	23.19	47.43	38.47
0.5	Sine	45.40	20.48	33.14	20.11
ARMA(1,1) Data with $\phi_1 = 0.8$ and $\theta_1 = -0.3$					
None		49.29	17.92	24.44	13.37
2.0	Ramp	48.12	17.85	23.97	13.41
2.0	Exp	48.99	18.29	22.94	13.10
2.0	Sine	48.18	18.05	24.06	14.15
1.5	Step	49.05	17.27	22.68	12.41
1.5	Ramp	46.48	19.65	26.51	16.93
1.5	Exp	48.32	17.86	23.45	13.90
1.5	Sine	46.40	19.05	25.68	15.44
1.0	Step	47.99	17.97	23.50	13.17
1.0	Ramp	44.86	19.98	28.48	18.81
1.0	Exp	42.77	20.36	31.03	21.98
1.0	Sine	43.03	17.88	29.08	16.59
0.5	Step	47.92	18.71	23.95	15.75
0.5	Ramp	40.77	22.33	40.37	34.86
0.5	Exp	41.02	23.04	41.88	36.90
0.5	Sine	46.05	20.39	30.69	19.52

gorithm does not perform as well on sequences that have slow shifts or quick large shifts from the transient to steady state. One possible explanation for this performance may be that the scalar \hat{k} is consistently estimated to be greater than 0.7. As the scalar \hat{k} nears one, the first element of the Kalman filter gain also nears one. The closer the scalar \hat{k} is to 1.0, the more the Kalman filter disregards the model state estimate and uses the last data for the best state estimate. If the scalar \hat{k} actually equals one, the predicted estimate is the last observations propagated by the state transition matrix. The result of a large Kalman filter gain is that the state estimates quickly adjust to changes in the measurement sequence. With a high Kalman filter gain, a jump discontinuity in the sequence is only apparent in the residual sequence as a single residual at the time of the discontinuity and of the magnitude of the jump shift. For this application, the filter's adjustment hides the effects of processing transient data. When transient data is processed, the filter adjusts because of the high gain, and the residuals remain relatively small. Without a significant change in the residuals, the cusum algorithm has no means to detect a change.

When applied to output of waiting times from a simulation of an $M/M/1$ queue, \hat{k} is estimated to be 1.0. Therefore, the algorithm adjusts to every shift in the output sequences. The residuals are just the difference between sequential values of the original output sequence. Thus, for that sample of simulation output, the residual monitoring algorithm is unsuccessful. One proposed solution is to run two filters, one with the estimated gain and another filter with a gain of zero. Perhaps by monitoring the rate of divergence of the filter with zero gain, the transient can be identified. Another potential approach may be to employ a bank of filters, like the MMAE parameter estimation technique, around the estimated steady-state mean. By monitoring the a posteriori probability of the filters' parameters, perhaps a good truncation point can be determined. This approach is developed and tested in the next chapter.

4.7 Summary

Data truncation is a commonly accepted method of dealing with initialization bias in discrete-event simulation. An algorithm for determining the appropriate initial-data truncation point for univariate output is studied. The technique entails beginning at the end of the simulation output and estimating a steady-state output model in a Kalman filter framework. Using the estimated model, the Kalman filter is applied and the residuals are monitored using a cumulative sums (cusum) quality control algorithm. The estimated truncation point is selected when the cusum algorithm indicates the Kalman filter residual sequence is "out-of-control".

A Monte Carlo analysis using data sets generated from AR(2) and ARMA(1,1) with induced transients is used to evaluate the technique. The performance measures include the ability to construct reliable confidence intervals for the mean response and passing Schruben's test for no initialization bias. The test cases selected probably have significantly more variability than actual discrete-event simulation output. Reasonable results are obtained for sequences with positive autocorrelations at lag one. Obviously, detecting small changes in the process mean is difficult.

The residual monitoring algorithm does not perform well in a limited test on actual simulation output. The simulation output is of waiting times for an $M/M/1$ queue. The parameter estimation step results in \hat{k} for 1.0. Thus the resulting Kalman filter adapted to every change in the output sequence with no significant pattern in the residuals. The technique is unable to detect the transient observations. The next chapter explores a different approach for the univariate single-run steady-state identification problem.

V. Single-Run Steady-State Identification: Multiple Model Adaptive Estimation (MMAE) Approach

5.1 Introduction

The truncation-point identification approach proposed in this chapter is based on Multiple Model Adaptive Estimation (MMAE). After estimating the steady-state Kalman filter parameters, the MMAE technique is applied to calculate a time-varying estimate of the output mean. A truncation point is selected when the MMAE mean estimate is "near" the steady-state mean estimate.

In the MMAE technique, the data sequence is processed through a bank of Kalman filters, each with a different set of parameters. After processing an observation, the probability of each filter having the correct parameters is calculated based on the filter residuals. The MMAE time-varying estimate of the parameters is calculated with a probabilistic weighted average of the filter parameters. MMAE is introduced and discussed in Section 3.4.2.

In this application, the steady-state Kalman filter parameters are estimated using the correlation technique [66]. The filters in the MMAE bank share the same steady-state parameter estimates, except they have different mean estimates. Starting at the beginning of the sequence, the data sequence is processed forward through the MMAE bank of filters. After processing each observation, the MMAE filter probabilities are updated and a new MMAE mean estimate based on all the previous data is calculated. Effective steady-state output may be detected in two ways. First, the filter probability for the filter using the steady-state mean estimate should be near one. Second, when the MMAE mean estimate is within a small tolerance of the steady-state mean estimate for a predetermined number of consecutive observations, the truncation point can be selected.

In related research, Howard [37] uses MMAE to construct confidence intervals for the univariate output of discrete-event simulations. He tests three different confidence interval techniques. The first technique is based on the estimated steady-state parameters and vari-

ances. The second confidence interval technique uses the MMAE estimate of the variance for the mean estimator, and the final technique is based upon the final MMAE filter probabilities. Since Howard uses the same Kalman filter model formulation, this analysis of MMAE characteristics provides some insight to Howard's confidence interval results.

Although the same Kalman filter formulation as used in Chapter IV is applied, a different parameter estimation scheme is used. After discussing parameter estimation, the MMAE truncation approach is developed, and a Monte Carlo analysis follows. This chapter is divided into the following major sections: parameter estimation, MMAE implementation, and Monte Carlo Analysis.

5.2 *Parameter Estimation*

In this section, the model formulation is reviewed. The parameter estimation with the correlation technique is developed. The accuracy of the variance estimates is analyzed.

In this chapter, the AR(2) with measurement noise formulation, from Section 4.2.1, is applied. Using the same system matrices shown in Equation (45), the simulation output y_n is modeled as the sum of a constant mean, AR(2) correlated noise, and independent white measurement noise. The four parameters that must be estimated are the steady-state mean μ_y , the autoregressive coefficients ϕ_1 and ϕ_2 , and the first element of the Kalman filter gain k . Based on estimates of these parameters and the sample residual mean, the dynamic noise variance Q_d , measurement noise variance R , and the Kalman filter state estimate variances P^- and P^+ can be estimated, if desired.

In Chapter IV, the four unknown parameters are estimated using least squares estimation. The data is processed from the end of the simulation output until the least squares estimates pass a heuristic criterion. In contrast, this approach, like Schruben's [87], assumes that the last half of the data sequence has effectively attained a stationary steady-state output distribution. Therefore, using the fixed sample size of half the number of data, the steady-state parameters are estimated.

Using Mehra's [66] correlation technique for parameter estimation, the first three theoretical lagged autocorrelations for an AR(2) with measurement noise model are equated with the sample autocorrelations. In this formulation, the AR(2) noise term $\tilde{\eta}(t_n)$ equals the first element of the state estimate, $\mathbf{x}_1(t_n^-) \equiv \tilde{\eta}(t_n)$. The state vector $\mathbf{x}(t_n)$ is a random vector with mean of $\hat{\mathbf{x}}(t_n)$ and variance of \mathbf{P}^- . Since the measurement noises are independent of the state estimates and independent in time, the lagged simulation output autocovariances are the same as for the AR(2) dynamics state estimate:

$$\gamma_y(1) = \gamma_{\tilde{\eta}}(1) = \frac{\phi_1}{1-\phi_2} \gamma_{\tilde{\eta}}(0)$$

$$\gamma_y(2) = \gamma_{\tilde{\eta}}(2) = \phi_1 \gamma_{\tilde{\eta}}(1) + \phi_2 \gamma_{\tilde{\eta}}(0) = \frac{\phi_1^2}{1-\phi_2} \gamma_{\tilde{\eta}}(0) + \phi_2 \gamma_{\tilde{\eta}}(0)$$

$$\gamma_y(3) = \gamma_{\tilde{\eta}}(3) = \phi_1 \gamma_{\tilde{\eta}}(2) + \phi_2 \gamma_{\tilde{\eta}}(1)$$

The variance $\gamma_{\tilde{\eta}}(0)$ equals the variance of the first element of the state estimate P_{11}^- . The variance of the simulation output, shown in Equation (55), can be expressed as the variance of the first element of the state estimate plus the measurement noise variance, $\gamma_y(0) = P_{11}^- + R$. The ratio $\frac{\gamma_{\tilde{\eta}}(0)}{\gamma_y(0)}$, or equivalently $\frac{P_{11}^-}{P_{11}^- + R}$, is seen to equal the unknown scalar k by Equation (48) on page 91. Dividing the autocovariances by the variance $\gamma_y(0)$ results in the theoretical autocorrelations $\rho_y(i)$. After substituting in the scalar k , the autocorrelation equations are

$$\begin{aligned} \rho_y(1) &= \frac{\phi_1}{1-\phi_2} k \\ \rho_y(2) &= \phi_1 \rho_y(1) + \phi_2 k \\ \rho_y(3) &= \phi_1 \rho_y(2) + \phi_2 \rho_y(1) \end{aligned} \tag{62}$$

Using these relationships, a quadratic equation for ϕ_1 can be derived. Solving the first two autocorrelations in Equation (62) for k results in

$$k = \frac{\rho_y(1) - \phi_2 \rho_y(1)}{\phi_1} = \frac{\rho_y(2) - \phi_1 \rho_y(1)}{\phi_2}$$

Cross multiplying and substituting in the relationship from the third autocorrelation,

$$\phi_2 = \frac{\rho_y(3) - \phi_1 \rho_y(2)}{\rho_y(1)} \text{ results in}$$

$$\begin{aligned} \phi_1 \rho_y(2) - \phi_1^2 \rho_y(1) &= \phi_2 \rho_y(1) - \phi_2^2 \rho_y(1) \\ &= \left(\frac{\rho_y(3) - \phi_1 \rho_y(2)}{\rho_y(1)} \right) \rho_y(1) - \left(\frac{\rho_y(3) - \phi_1 \rho_y(2)}{\rho_y(1)} \right)^2 \rho_y(1) \\ &= \rho_y(3) - \phi_1 \rho_y(2) - \frac{(\rho_y^2(3) - 2\phi_1 \rho_y(2) \rho_y(3) + \phi_1^2 \rho_y^2(2))}{\rho_y(1)} \end{aligned}$$

All the terms are moved to one side of the equation and grouped terms in powers of ϕ_1 :

$$\rho_y(1) \phi_1^2 - \frac{\rho_y^2(2)}{\rho_y(1)} \phi_1^2 + \frac{2\rho_y(2)\rho_y(3)}{\rho_y(1)} \phi_1 - 2\rho_y(2) \phi_1 + \rho_y(3) - \frac{\rho_y^2(3)}{\rho_y(1)} = 0$$

The actual parameter estimates for $\hat{\phi}_1$, $\hat{\phi}_2$, and \hat{k} are calculated using the sample autocorrelations. Using the last half of the output sequence, the steady-state output mean μ_y is estimated with the sample mean,

$$\bar{y} = \frac{2}{N-2} \sum_{n=\frac{N}{2}+1}^N y_n \quad (63)$$

The sample autocorrelations $r_y(i)$ at lag i are calculated with

$$r_y(i) = \frac{\sum_{n=\frac{N}{2}+1}^{N-i} (y_n - \bar{y})(y_{n+i} - \bar{y})}{\sum_{n=\frac{N}{2}+1}^N (y_n - \bar{y})^2} \quad (64)$$

Substituting the sample autocorrelations, a quadratic equation in terms of $\hat{\phi}_1$ is determined:

$$\left(r_y(1) - \frac{r_y^2(2)}{r_y(1)} \right) \hat{\phi}_1^2 + \left(\frac{2r_y(2)r_y(3)}{r_y(1)} - 2r_y(2) \right) \hat{\phi}_1 + \left(r_y(3) - \frac{r_y^2(3)}{r_y(1)} \right) = 0 \quad (65)$$

Using the solutions for a quadratic equation, two estimates of $\hat{\phi}_1$ may be available. For each quadratic solution for $\hat{\phi}_1$, the third autocorrelation equation is used to estimate the second autoregressive coefficient, $\hat{\phi}_2 = \frac{r_y(3) - \hat{\phi}_1 r_y(2)}{r_y(1)}$. In an effort to minimize the numerical error, the

Table 52. Correlation Technique Parameter Estimation Summary

μ_y \bar{y}			ϕ_1 $\hat{\phi}_1$			ϕ_2 $\hat{\phi}_2$			k \hat{k}		
$\hat{\mu}_y$ $\hat{\sigma}_y$			$\hat{\mu}_{\hat{\phi}_1}$ $\hat{\sigma}_{\hat{\phi}_1}$			$\hat{\mu}_{\hat{\phi}_2}$ $\hat{\sigma}_{\hat{\phi}_2}$			$\hat{\mu}_{\hat{k}}$ $\hat{\sigma}_{\hat{k}}$		
10.00	10.00	0.32	0.60	0.63	0.12	0.30	0.27	0.11	0.57	0.80	0.05

pair of autoregressive parameters $(\hat{\phi}_1, \hat{\phi}_2)$ which minimizes $|r_y(3) - \hat{\phi}_1 r_y(2) - \hat{\phi}_2 r_y(1)|$ from the third autocorrelation equation is selected. (In a test of 1000 runs, the truncation algorithm using the other pair of estimates obtained very similar, but slightly inferior, results.) Using these estimates for the autoregressive parameters, $\hat{k} = \frac{(1 - \hat{\phi}_2)r_y(1)}{\hat{\phi}_1}$ from the autocorrelation equation for lag one. The resulting estimates of $\hat{\mu}_y$, $\hat{\phi}_1$, $\hat{\phi}_2$ and \hat{k} are the correlation technique estimates.

Based on the estimates, the gain and variances can be determined. Using Equation (51) the Kalman filter gain K is calculated. The estimation of the noise and state variances (\hat{Q}_d , \hat{R} , P^- , and P^+) is the same as described in Section 4.3.3. However, the bias in the estimation of \hat{k} affects the variance estimates, as demonstrated below.

As a test of this estimation scheme, this correlation technique is applied to 3,000 generated data sequences. The model for the generated data is an AR(2) with measurement noise with autoregressive coefficients of $\phi_1 = 0.6$ and $\phi_2 = 0.3$, dynamics noise variance $Q_d = 1.0$, and measurement noise variance $R = 1.0$. For each generated sequence, the parameter estimates are based on 1,000 steady state observations. As seen in Table 52, the estimates of the mean and the autoregressive coefficients are fairly accurate; however, the estimate of the scalar \hat{k} is significantly in error. Since the value of the estimate \hat{k} is integral to the estimation of the noise variances (See Section 4.3.3), these calculations are completed twice; once with the estimated scalar \hat{k} and again with the true k . Table 53 shows that the biased estimate \hat{k} induces significant error into the noise variance estimates \hat{Q}_d and \hat{R} . However, if \hat{k} can be accurately estimated, the noise variance estimates are also accurate.

Most estimation schemes for Kalman filter parameters, such as MLE, MMAE, and

Table 53. Variance Estimation Summary

	Q_d			R			σ_y		
	\hat{Q}_d	$\hat{\mu}_{\hat{Q}_d}$	$\hat{\sigma}_{\hat{Q}_d}$	\hat{R}	$\hat{\mu}_{\hat{R}}$	$\hat{\sigma}_{\hat{R}}$	$\hat{\sigma}_y$	$\hat{\mu}_{\hat{\sigma}_y}$	$\hat{\sigma}_{\hat{\sigma}_y}$
Estimated \hat{k}	1.00	1.68	0.18	1.00	0.48	0.13	2.27	2.80	0.39
True k	1.00	1.01	0.08	1.00	1.00	0.05	2.27	2.33	0.27

Table 54. Sum of Squared Residuals for Various Scalar \hat{k}

\hat{k}	Estimated		True	
	$\hat{\mu}_y, \hat{\phi}_1, \hat{\phi}_2$		μ_y, ϕ_1, ϕ_2	
	$\sum_{n=1}^{20,000} r_n^2$	% Change	$\sum_{n=1}^{20,000} r_n^2$	% Change
0.0	104,472.8	124.6	104,729.6	125.2
0.1	65,545.0	40.9	64,423.2	38.5
0.2	54,414.0	17.0	53,718.6	15.5
0.3	49,763.7	7.0	49,355.6	6.1
0.4	47,620.5	2.4	47,397.2	1.9
0.5	46,711.8	0.4	46,615.6	0.2
0.6	46,506.4	0.0	46,502.8	0.0
0.7	46,744.9	0.5	46,812.8	0.7
0.8	47,286.4	1.7	47,411.9	2.0
0.9	48,047.6	3.3	48,221.6	3.7
1.0	48,978.6	5.3	49,194.9	5.8

Note: True $k = 0.57$.

least squares, are based on functions of the squared residuals. In an effort to assess the ambiguity in the estimation of the scalar k , one sequence of 20,000 steady-state observations is examined. The sequence was generated with an AR(2) with measurement noise model. The sum of squared residuals are calculated for each value of the scalar \hat{k} varied at increments of 0.1 over its feasible range of zero to one. The first set of columns for the sum of squared residuals in Table 54 is calculated with the parameter estimates of $\hat{\mu}_y = 10.15$ ($\mu_y = 10.0$), $\hat{\phi}_1 = 0.595$ ($\phi_1 = 0.6$), and $\hat{\phi}_2 = 0.291$ ($\phi_2 = 0.3$). Even with these relatively accurate parameter estimates, the effect of the scalar \hat{k} is a very small percentage change in the sum of 20,000 residuals over most of the feasible region. To ensure that the estimation errors of the

other parameters are not reducing the effect of the variation in \hat{k} , the test is conducted again using the true parameters except for \hat{k} . Even when fixing the parameters which generated the data to their true values, the effect of varying \hat{k} is minimal for most of the feasible range. The length of this data sequence is ten times greater than the simulation output sequences tested in this research. Thus, the value of \hat{k} may not be estimated accurately with a limited number of observations available in many simulation applications.

Howard [37] in his confidence interval techniques uses a slightly different approach to estimating the parameters. He uses the first two correlations to estimate the autoregressive coefficients assuming a value for the scalar \hat{k} . A linear search for \hat{k} between its bounds of zero and one is applied to minimize the sum of squared residuals. Howard applies the same equations to calculate the variance as the results in Table 53. In his tests, the scalar \hat{k} is overestimated. Thus, the dynamics noise variance generally is overestimated and the measurement noise variance is underestimated. These biased variance estimates are the basis of the first confidence interval technique Howard tested. Probably because of the bias in the estimation of \hat{k} , the confidence interval technique does not perform as well as other techniques.

Table 54 indicates that the value of \hat{k} has a minor impact on the sum of squared residuals for this very long sequence. The implication is that \hat{k} may not be estimated accurately with a limited number of observations. In the opposite view, the value of \hat{k} does not significantly degrade the performance of the Kalman filter. Perhaps the value of \hat{k} may be arbitrarily set to achieve desired filter performance characteristics.

Using the above estimation scheme based on the correlation technique, estimates of the mean, the autoregressive coefficients, and the scalar \hat{k} are obtained with only two passes through the data sequence. Using these estimated parameters, the last half of the data sequence is processed through the estimated steady-state Kalman filter. The mean residual

squared error is used as an estimate of the residual variance:

$$\mathbf{H}\mathbf{P}^-\mathbf{H}^T + R = \frac{2}{N} \sum_{n=\frac{N}{2}}^N r_n^2 \quad (66)$$

With these estimates, a bank of Kalman filters can be set up, and MMAE may be applied.

5.3 Multiple Model Adaptive Estimation (MMAE) Implementation

This section describes the application of MMAE to the problem of steady-state identification. The implementation issues, particularly the appropriate magnitude of the scalar k , are discussed. The section identifies two bases for determining that steady-state observations are being processed.

MMAE is introduced in Section 3.4.2 as a parameter estimation technique. In MMAE, a bank of Kalman filters, each with a different set of parameters, is run through the data sequence. After processing each observation, the probability of each filter having the correct parameters is calculated with Equation (36). Using the filter probabilities, an MMAE estimate of the unknown parameters is determined with Equation (39) with conditional covariance matrix of the estimated parameters calculated with Equation (40).

In this application, the steady-state system parameter estimates, calculated with the correlation technique, are used in the MMAE bank of Kalman filters. However, each filter is based on a different mean estimate. Thus, the MMAE filter probabilities and estimates give indication to changes in the mean of the data sequence.

Application of the MMAE approach involves determination of the following:

- Number and spacing of parameter estimates in the MMAE bank
- Value of the Kalman filter scalar \hat{k}
- Filter prior probabilities, $p_l(t_0)$ in Equation (36)
- Determining if and when to reset the filter probabilities $p_l(t_n)$

- Establishing rules to declare steady-state by selecting \hat{n}_0

As Howard [37, 39] reports, these issues can be critical to successfully applying MMAE to simulation output. The issues are difficult to resolve since their effects interact in the performance of the MMAE technique. Each of these issues is addressed in the following sections.

5.3.1 Number and Spacing of Filters. While in some MMAE applications many filters are incorporated into the bank, the filters in this application are intended to signal three possibilities about the output sequence: a downward bias, an upward bias, or no bias. The objective is to truncate the initial biased data that significantly affect the mean estimator. Therefore, in this application, detecting biases of different sizes is important, but not estimating the magnitude of a bias. The selected bank consists of three filters, one centered on the steady-state mean estimate and the others spaced above and below.

The spread between the filter means also influences the MMAE filter probabilities, and hence the MMAE mean estimates. Let the steady-state mean based on the last half of the data be represented by \bar{y} . The mean estimates in the three filters are $\hat{\mu}_j = \bar{y} + \delta_j$ where δ_j is negative for the low filter, zero for the center filter, and positive for the high one. The effect of spacing is examined with \hat{k} set equal to zero and equal to one. These values evaluate the possible range of the filter gains. $\hat{k} = 0$ means the Kalman filter relies only on the dynamics model and disregards measurement updates. In contrast, $\hat{k} = 1.0$ means that the state estimates are based on the propagation of the last two measurements. In both cases, the filter parameters need to be spaced far enough apart to differentiate between the filters' residuals. The actual spacing necessary is a function of the variation in the simulation output being processed.

First, the Kalman filter residuals are examined with no measurement updates. If $\hat{k} = 0$, the Kalman filter gain $\mathbf{K} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ by Equation (51). Assuming stationary estimates for AR(2) coefficients $\hat{\phi}_1$ and $\hat{\phi}_2$, without measurement updates, the Kalman filter state

estimates $\hat{\mathbf{x}}(t_n^-)$ and $\hat{\mathbf{x}}(t_n^+)$ decay to zero vectors. For each filter, the measurement z_n^j is the data y_n minus that filter's mean μ_y^j . For each of three filters, the residual at time t_n is

$$\begin{aligned} r_n^j &= z_n^j - \mathbf{H}\hat{\mathbf{x}}(t_n^-) \\ &= y_n - \mu_y^j \\ &= y_n - \bar{y} - \delta_j \end{aligned}$$

The data y_n and the estimated steady-state mean \bar{y} are the same for each filter. Each filter's residuals are determined by the variation in y_n and the size of δ_j . Since the relative magnitudes of the filter residuals determine the MMAE filter probabilities, the size of δ_j compared to the variation in y_n may affect the stability of the filter probabilities. Relatively stable filter probabilities make the MMAE mean estimate smoothly progress as more observations are processed. Therefore δ_j should probably be large compared to σ_y .

Similarly, the MMAE filter parameters should be widely spaced when the Kalman filters' estimates rely extensively on the measurements. For $\hat{k} = 1$, the Kalman filter gain $\mathbf{K} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ by Equation (51). From Equations (22), (25) and (45), the corresponding state estimates for the j th filter are $\hat{\mathbf{x}}_j(t_{n-1}^+) = \begin{bmatrix} z_{n-1}^j \\ z_{n-2}^j \end{bmatrix}$ and $\hat{\mathbf{x}}(t_n^-) = \begin{bmatrix} \hat{\phi}_1 z_{n-1}^j + \hat{\phi}_2 z_{n-2}^j \\ z_{n-1}^j \end{bmatrix}$. The residuals for each of the filter are

$$\begin{aligned} r_n^j &= z_n^j - \mathbf{H}\hat{\mathbf{x}}_j(t_n^-) \\ &= z_n^j - \hat{\phi}_1 z_{n-1}^j - \hat{\phi}_2 z_{n-2}^j \\ &= (y_n - \mu_y^j) - \hat{\phi}_1 (y_{n-1} - \mu_y^j) - \hat{\phi}_2 (y_{n-2} - \mu_y^j) \\ &= (y_n - \bar{y} - \delta_j) - \hat{\phi}_1 (y_{n-1} - \bar{y} - \delta_j) - \hat{\phi}_2 (y_{n-2} - \bar{y} - \delta_j) \\ &= (y_n - \bar{y}) - \hat{\phi}_1 (y_{n-1} - \bar{y}) - \hat{\phi}_2 (y_{n-2} - \bar{y}) - (1 - \hat{\phi}_1 - \hat{\phi}_2)\delta_j \end{aligned}$$

The last three observations minus the steady-state mean estimate are the same for each filter. For stationary AR(2) processes, the factor $(1 - \hat{\phi}_1 - \hat{\phi}_2)$ is bounded between 0 and 4. With $\hat{k} = 1$, the spacing between filters δ_j must be sufficiently large to distinguish between

the different filters' residuals.

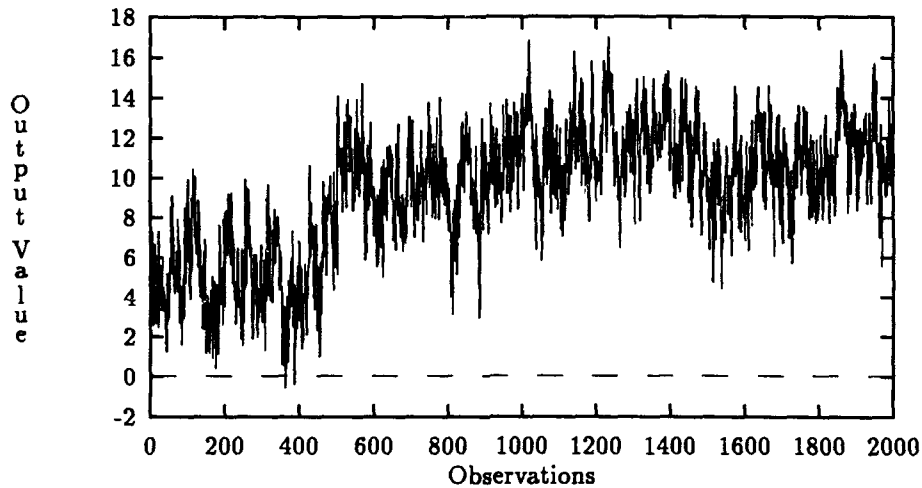
Whether $\hat{k} = 0$ or $\hat{k} = 1$, the differences between the filter means need to be sufficiently large to result in significant difference in the filter residuals, and hence, the MMAE filter probabilities. If the filter means are too close together, the variation in the output sequence causes the probability to fluctuate between filters. If the filters are spread too far apart, regardless of the transient, the significant probability remains with the center filter.

In a Monte Carlo analysis, two rules for determining the filter spread are considered. Initially, the filters are spread based on the variation in the filter residuals. Using the estimated steady-state parameters and the last half of the output sequence, the residual mean square error is calculated, $\mathbf{HP}^{-1}\mathbf{H}^T + R = \frac{2}{N-2} \sum_{n=\frac{N}{2}+1}^N r_n^2$. The filter spread of $\delta_j = 3[\mathbf{HP}^{-1}\mathbf{H}^T + R]^{\frac{1}{2}}$ is tested initially. However, the filter parameter range that worked for the generated time-series models is not adequate with simulation output of an $M/M/1$ queue. The magnitude of the residuals can vary depending on how well an AR(2) with measurement noise model can fit the output sequence. Since the different filter means are used to detect changes in the output sequence, the variation of the output, rather than the residuals, seems to provide a better indication of the appropriate filter spacing. Since no reliable variance estimate for the output sequence, σ_y^2 , is readily available, the filters are spread based on the minimum and maximum data values in the last half of the output sequence. The lower and upper filters are positioned at 90 percent of the distance between the estimated mean and the respective extreme data values.

In this subsection, the number of filters for the MMAE bank is chosen to be three. The filters' means are based on the estimated mean and either the residual standard deviation or the extreme simulation output values. The next subsection investigates the effect of varying the magnitude of the Kalman filter gain.

5.3.2 Kalman Filter Gain. In this subsection, the effect on the MMAE estimates of the Kalman filter gains of different magnitudes are examined. The investigation has two purposes. The first is to select the value of \hat{k} , which determines the magnitude of the

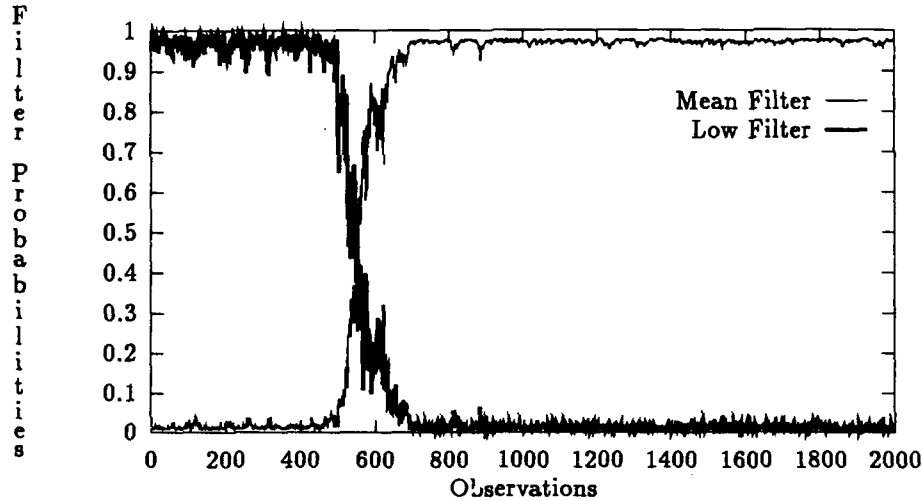
Figure 13. Sample AR(2) with Measurement Noise Data Sequence



Kalman filter gain, that results in good performance of the MMAE technique. The second objective is to identify MMAE estimates that with an associated criteria may constitute a good truncation-point selection algorithm. The MMAE filter probabilities, state estimates, residuals, MMAE mean estimates and variances are considered. The MMAE filter probabilities and MMAE mean estimates are selected as potential indicators for a truncation-point selection algorithm.

In investigating the effects of various Kalman filter gains, one data sequence is used extensively. The sequence, shown in Figure 13, was generated from an AR(2) with measurement noise model using the following parameters; autoregressive coefficients $\phi_1 = 0.6$, $\phi_2 = 0.3$, dynamics noise variance $Q_d = 1.0$ and measurement noise variance $R = 1.0$. The sequence consisted of 2,000 observations. The last 1500 observations are steady-state values with a mean of 10.0 and a standard deviation $\sigma_y = 2.37$. The first 500 observations have a transient mean of $\mu_y - 2\sigma_y = 5.46$. The transition from the transient to steady-state is a step change in the mean at the 500th observation. While this is an artificial transient, the large step change should be clearly identifiable by potential measures for selecting the appropriate

Figure 14. MMAE Filter Probabilities with $\hat{k} = 1.0$



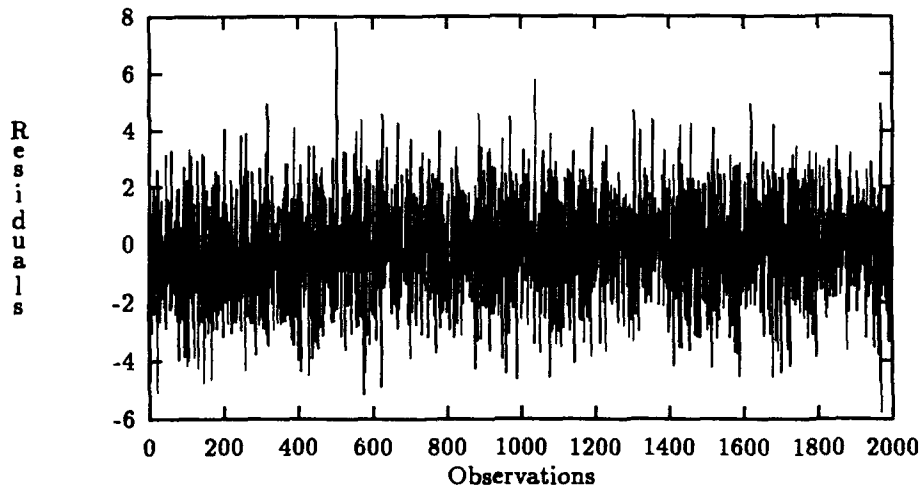
truncation point. In fact, the transient is obvious in Figure 13, whereas transients of actual simulation output often are not visibly discernible.

Applying the parameter estimation scheme to this sequence results in the following estimates: $\bar{y} = 11.17$, $\hat{\phi}_1 = 0.88$, $\hat{\phi}_2 = 0.04$, $\hat{k} = 0.65$, and the standard deviation of the residuals is $[\mathbf{HP} - \mathbf{H}^T + \mathbf{R}]^{\frac{1}{2}} = 1.55$. In these initial tests, the filter spacing is based on plus and minus three times the standard deviation of the steady-state filter residuals. The three MMAE filter means are 6.52, 11.17, and 15.82.

After applying MMAE, the filter probabilities are examined. Figure 14 shows the a posteriori probabilities of the filter parameters. One potential alternative for steady-state identification is to declare a truncation point when the probability of the filter based on the mean estimate \bar{y} achieves a high value. With $\hat{k} = 1.0$, the filter probabilities smoothly and quickly transitioned after the sequence changes to steady state at the 500th observation.

In Figure 15, the residuals for the center filter, which used the steady-state mean estimate \bar{y} , are shown. The step change in the mean at observation 500 is apparent as a single

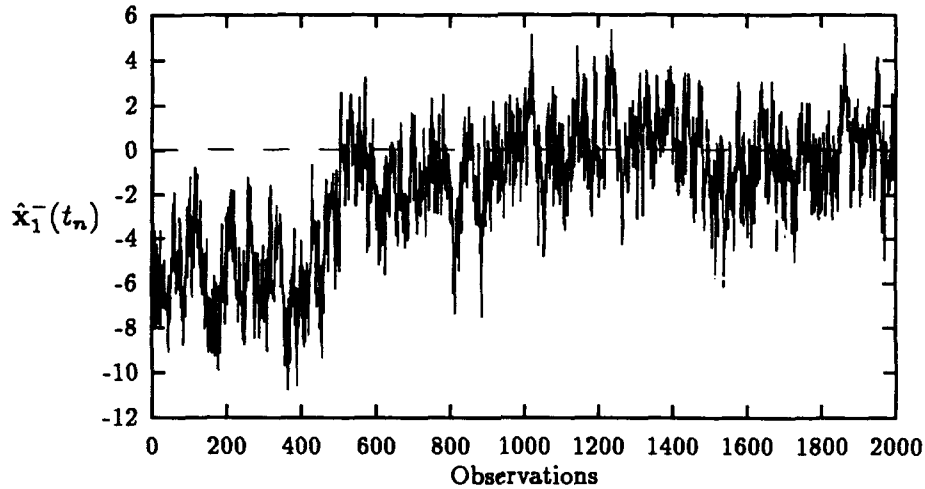
Figure 15. Residuals of Mean Filter with $\hat{k} = 1.0$



large residual. Only one large residual resulted because with $\hat{k} = 1.0$, the state estimates adapted in one iteration to changes in the data sequence. The one large residual occurs only for transients that end with a step change to steady state. Transients that decay gradually, which is common in discrete-event simulation output, produce no significant pattern in the residuals because the state estimates adapt to every small change in the transient. In Chapter IV, the residuals are monitored in an attempt to detect the change between the transient and steady-state. Because the first element of the estimated Kalman filter gain \hat{k} is generally very near one, the state estimates adapt rapidly to any change in the data. Therefore, the residual sequence have little or no pattern that can be detected. Even with smaller gains, which are attempted later, the residuals contain no easily identifiable pattern to detect a gradual change. Therefore, tracking the residuals is not considered further as a viable basis for a truncation-point selection approach.

The state estimates for a particular filter, $\hat{x}_1(t_n^-)$, do indicate a change in the output sequence. The state estimates $\hat{x}_1(t_n^-)$ for this sequence are shown in Figure 16. However, the sequence of state estimates is correlated. In fact, with $\hat{k} = 1.0$, the state estimates are simply

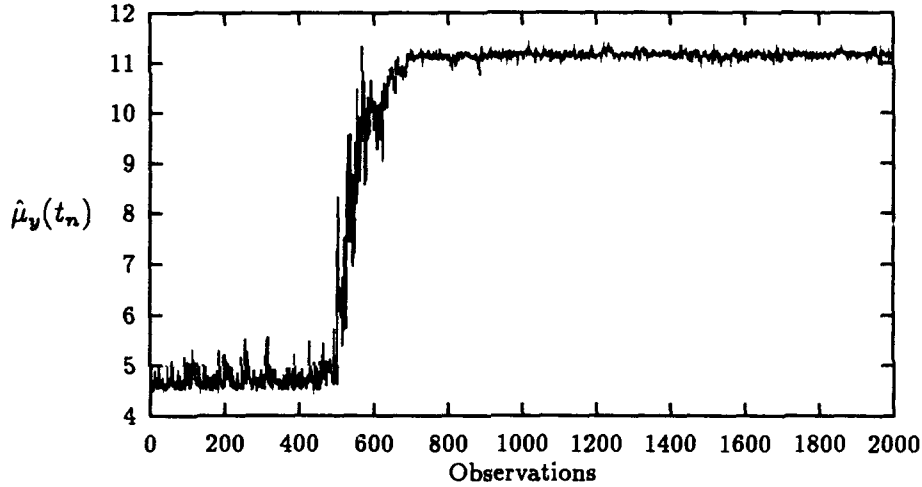
Figure 16. $H\hat{x}^-(t_n)$ of Mean Filter with $\hat{k} = 1.0$



the original sequence corrected for the filter's mean estimate. With lower values of \hat{k} , the state estimates decrease in correlation, but the transient also becomes less obvious. Because of the correlation, detecting the change in state estimates is no easier than finding the change in the original data sequence. The MMAE state estimates, determined by a weighted average of the filter state estimates and their respective probabilities, has no discernible pattern. Therefore, the MMAE state estimates are not pursued further as a basis for identifying steady state.

Besides the individual filter state estimates and residuals, the MMAE technique also provides an MMAE mean estimator and variance of the mean estimator. Figure 17 shows the MMAE mean estimates $\hat{\mu}_y(t_n)$, which are the filter means weighted by the filter probabilities. With $\hat{k} = 1.0$ and the bank of three filters as specified, the MMAE mean quickly detects a large step change in the mean. The assumed steady-state mean estimate \bar{y} is 11.1, so the center filter is positioned there. After 500 transient observations, the MMAE mean estimate $\hat{\mu}_y(t_n)$ stabilizes on the center filter's mean estimate. Because of the clear indication of the

Figure 17. MMAE $\hat{\mu}_y(t_n)$ with $\hat{k} = 1.0$



transition to steady state, the MMAE mean estimator is the basis of the second truncation-point selection algorithm tested in a Monte Carlo analysis.

Figure 18 depicts the MMAE conditional standard deviation of the mean estimate $\hat{\sigma}_{\hat{\mu}_y}(t_n)$ calculated with Equation (40) where $\hat{a}(t_n) = \hat{\mu}_y(t_n)$. The second confidence interval technique tested by Howard [37] is based on the final MMAE variance estimate. Because of the variability in these estimates as each data point is processed, that confidence interval technique may be improved slightly by using an average of a group of estimates at the end of the data.

The MMAE technique provides at least two potential indicators of steady state. The first is that the MMAE filter probability for the filter using the mean estimate of \bar{y} approaches one. The second indicator is that the MMAE time-varying mean estimate approaches \bar{y} . The following analysis investigates the effect of changing the magnitude of the Kalman filter gain on these two indicators.

Using the same data sequence, MMAE is applied with the Kalman filter gain scalar

Figure 18. MMAE $\hat{\sigma}_{\hat{\mu}_y}(t_n)$ with $\hat{k} = 1.0$

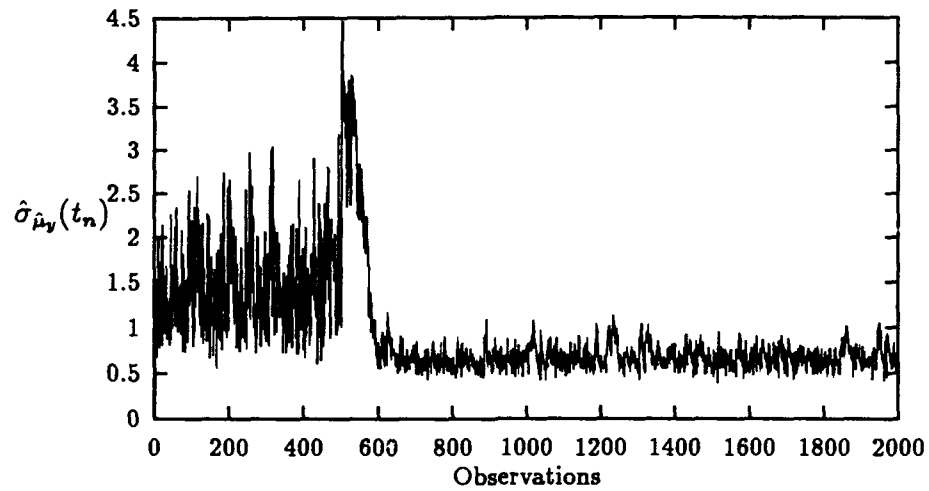
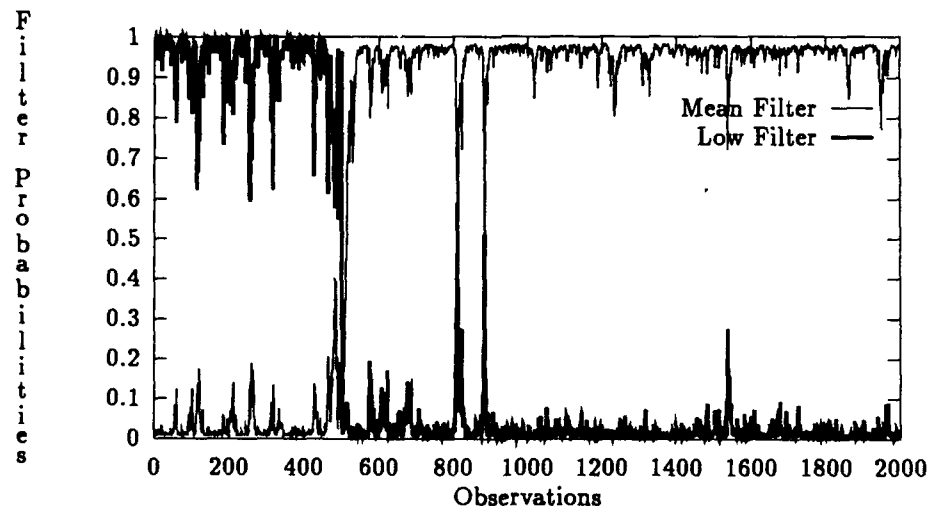
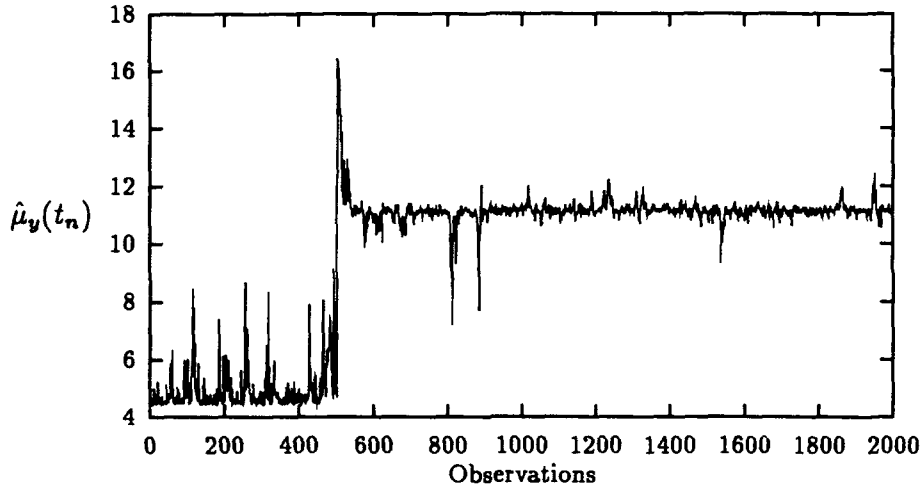


Figure 19. MMAE Filter Probabilities with $\hat{k} = 0.5$



\hat{k} set equal to 0.5. Figure 19 shows the MMAE filter probabilities for the filters. With a lower value for \hat{k} , the filter probabilities are more sensitive to changes in the data sequence.

Figure 20. MMAE $\hat{\mu}_y(t_n)$ with $\hat{k} = 0.5$



However, the probabilities continue to fluctuate even when steady-state observations are being processed. Similar fluctuations occur in the MMAE estimate of the mean $\hat{\mu}_y(t_n)$, shown in Figure 20.

The value of \hat{k} selected determines the performance of the MMAE filter probabilities $p_j(t_n)$ from Equation (36). When the scalar \hat{k} is set at or near its upper bound of one, the MMAE filter probabilities are very stable. As \hat{k} decreases, the probabilities, and hence the MMAE mean estimates $\hat{\mu}_y(t_n)$, increase in fluctuations. However, with a lower \hat{k} , the probabilities and the MMAE mean estimate respond faster to a change in the sequence. By varying \hat{k} , a tradeoff can be made between stable consistent indicators and faster indicators with more false signals. Because with the higher \hat{k} the indicators are slower to respond, larger values of \hat{k} also result in selecting truncation points that eliminate more of the initial data. Since $\hat{k} = 1.0$ results in clearer indications and conservative truncation points, $\hat{k} = 1.0$ is implemented in the Monte Carlo analysis in this chapter. In Chapter VI, the same truncation-point selection method is applied to the average of multiple replications. In those applications, better truncation points are selected when the estimated \hat{k} is used in the

algorithm rather than arbitrarily setting \hat{k} to one.

5.3.3 Filter Probabilities. To begin the recursive MMAE procedure, filter a priori probabilities for time t_0 have to be set. In other applications of MMAE, the prior probabilities $p_j(t_0)$ are significant [37]. However, for the test data sequence, the effect of the priors is insignificant after processing 40 observations. Therefore, the priors are set uniformly with each of the three filters having an a priori probability of a third; $p_j(t_0) = \frac{1}{3}$.

Each filter probability represents the probability conditioned on all data available to that time. Thus, the filter probabilities are affected by the transient data. To eliminate the effect of the transient, the probabilities can be periodically reset to their prior probabilities. Resetting the filter probabilities was tested, but was found to have little effect. With dispersed filter spacing, the probabilities adapt rapidly to changes in the sequence without resetting. The decaying effect of the previous data is consistent with the decreasing significance of the filter probability priors as more data is processed.

5.3.4 Declaring the Transient: Heuristic Rules. In order to select the initial data truncation point \hat{n}_0 , two different heuristics are proposed and tested. The first approach simply selects \hat{n}_0 at the point when the probability of the mean filter exceeds 0.9. The second approach monitors the MMAE mean estimate $\hat{\mu}_y(t_n)$. Welch's [101] graphical truncation-point identification scheme, described in Section 2.4.2, estimates the time-varying mean by smoothing the average of multiple simulation replications. Using Welch's method, the analyst selects the truncation point where the mean estimates stabilize. In a similar way, the second proposed method selects the truncation point when the MMAE time-varying mean estimate $\hat{\mu}_y(t_n)$ stabilizes within a small tolerance of the steady-state mean estimate, \bar{y} . When the absolute difference between the mean estimates, $ABS(\hat{\mu}_y(t_n) - \bar{y})$, is less than a small tolerance for a few consecutive observations, the truncation point \hat{n}_0 is selected. For these tests, the small tolerance is set to 1.0 and the criterion is passed for 5 consecutive observations. Both truncation-point selection methods performed well for generated transients, as seen in the next section.

5.4 Monte Carlo Results

A Monte Carlo analysis to test the effectiveness of using the MMAE approach to determine data truncation points is conducted. Two types of data are used. The first set of sequences are AR(2) with measurement noise data. These time-series sequences use the same system parameters as the sample sequence in Figure 13 with various types of transients. The second test set consists of waiting times from simulations of an $M/M/1$ queue. The truncation-point selection methods perform well on the time-series data, but not as successfully on the simulation output. The output from a single simulation may be a very difficult transient to detect.

5.4.1 AR(2) with Measurement Noise Data. The parameters used to generate the data are $\mu_y = 10.0$, $\phi_1 = 0.6$, $\phi_2 = 0.3$, $Q_d = 1.0$ and $R = 1.0$. Each sequence consists of 500 transient observations and 1,500 steady-state observations. Four types of transients; step, ramp, decaying exponential, and exponentially decaying sinusoid, are used. The magnitude of the transient shift $\Delta_y(\sigma_y)$ at the 250th observation is varied in multiples of the standard deviation of the steady-state data, $\sigma_y = 2.27$. The transients are shown in Figure 12 on page 116.

For both truncation-point selection rules, the parameter estimation procedure is the same and the MMAE implementation is identical. The MMAE filters are positioned at the mean and $\pm 3 [\mathbf{HP}^{-1}\mathbf{H}^T + R]^{\frac{1}{2}}$. The estimated system parameters are shown in Table 55. The scalar \hat{k} is arbitrarily set to one in this Monte Carlo analysis.

The simple rule of declaring the steady state when the mean filter had an MMAE probability greater than ninety percent, $\text{Prob}(\hat{\mu}_y(t_n)) > 0.9$, is implemented. The truncation-

Table 55. Correlation Technique Parameter Estimation Summary

μ_y	\bar{y}		ϕ_1	$\hat{\phi}_1$		ϕ_2	$\hat{\phi}_2$		\hat{k}	\hat{k}	
	$\hat{\mu}_{\bar{y}}$	$\hat{\sigma}_{\bar{y}}$		$\hat{\mu}_{\hat{\phi}_1}$	$\hat{\sigma}_{\hat{\phi}_1}$		$\hat{\mu}_{\hat{\phi}_2}$	$\hat{\sigma}_{\hat{\phi}_2}$		$\hat{\mu}_{\hat{k}}$	$\hat{\sigma}_{\hat{k}}$
10.00	10.00	0.32	0.60	0.63	0.12	0.30	0.27	0.11	0.57	0.80	0.05

Table 56. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$) ($\text{Prob}(\hat{\mu}_y(t_n)) > 0.9$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = 0.6$ and $\phi_2 = 0.3$							
2.0	Step	0.00	-498	1406	177.8	163.0	58170.3
2.0	Ramp	0.00	-212	1075	89.0	147.2	29589.1
2.0	Exp	0.00	-184	1146	149.1	136.1	40754.2
2.0	Sine	0.00	-498	123	-390.5	161.9	178696.9
1.5	Step	0.00	-498	886	75.8	250.9	68704.4
1.5	Ramp	0.00	-259	1213	67.9	153.7	28235.0
1.5	Exp	0.00	-289	1093	102.1	168.5	38802.7
1.5	Sine	0.00	-498	226	-410.4	136.4	186985.9
1.0	Step	0.00	-498	924	-202.0	296.5	128721.8
1.0	Ramp	0.00	-498	868	-18.7	166.2	27973.6
1.0	Exp	0.00	-306	995	8.8	161.3	26083.1
1.0	Sine	0.00	-498	339	-422.1	118.5	192186.5
0.5	Step	0.00	-498	925	-438.9	123.6	207922.1
0.5	Ramp	0.00	-498	824	-309.5	195.4	133990.1
0.5	Exp	0.00	-383	1135	-85.2	154.6	31145.5
0.5	Sine	0.00	-498	3	-433.6	101.3	198285.4

Note: Negative Truncation points indicate transient observations are retained.

point errors for 1,000 runs are summarized in Table 56. In this table, "Fail" indicates the rate of times that no truncation point is selected. Since the "Fail" column is all zeros, the algorithm always selected a truncation point.

The desired truncation-point error is a small positive number so that all of the transient data is truncated and little excessive truncation of steady-state data occurs. Large negative errors indicate that many transient observations are retained in the truncated sequence and they may bias estimates of steady-state parameters. Large positive truncation-point errors indicate that many steady-state observations are truncated, increasing the variability of steady-state parameter estimates.

The truncation-point algorithm based on the mean filter's probability generally works well. The transients with larger shifts, with the exception of the "Sine", results in small positive truncation-point errors. The transient of an exponentially decaying sinusoid, shown

Table 57. AR(2) Truncation Point \hat{n}_o Evaluations ($\text{Prob}(\hat{\mu}_y(t_n)) > 0.9$)

Transient		Fail	NOBM		OBM		Bias Test	
$\Delta\mu_y(\sigma_y)$	Type		Cov	Half Width	Cov	Half Width	$F(\hat{\sigma}_y^2)$	$\chi^2(\sigma_y^2)$
AR(2) Data with $\phi_1 = 0.6$ and $\phi_2 = 0.3$								
	None	0.02	0.85	0.392 ± 0.146	0.84	0.376 ± 0.124	0.98	0.93
2.0	Step	0.05	0.83	0.414 ± 0.167	0.82	0.403 ± 0.153	0.96	0.93
2.0	Ramp	0.03	0.86	0.405 ± 0.163	0.85	0.391 ± 0.146	0.99	0.95
2.0	Exp	0.03	0.82	0.408 ± 0.157	0.82	0.395 ± 0.136	0.97	0.94
2.0	Sine	0.03	0.92	0.448 ± 0.141	0.90	0.423 ± 0.101	0.63	0.47
1.5	Step	0.15	0.85	0.412 ± 0.169	0.82	0.399 ± 0.150	0.85	0.82
1.5	Ramp	0.02	0.83	0.399 ± 0.159	0.82	0.386 ± 0.141	0.98	0.94
1.5	Exp	0.03	0.84	0.409 ± 0.167	0.82	0.394 ± 0.147	0.96	0.93
1.5	Sine	0.02	0.90	0.445 ± 0.162	0.88	0.398 ± 0.090	0.66	0.48
1.0	Step	0.35	0.69	0.439 ± 0.173	0.67	0.418 ± 0.152	0.57	0.43
1.0	Ramp	0.04	0.83	0.392 ± 0.147	0.81	0.377 ± 0.130	0.94	0.91
1.0	Exp	0.02	0.85	0.396 ± 0.150	0.84	0.382 ± 0.136	0.98	0.95
1.0	Sine	0.02	0.91	0.441 ± 0.157	0.87	0.382 ± 0.087	0.71	0.54
0.5	Step	0.15	0.67	0.384 ± 0.134	0.66	0.367 ± 0.119	0.77	0.55
0.5	Ramp	0.10	0.77	0.387 ± 0.128	0.75	0.369 ± 0.110	0.85	0.70
0.5	Exp	0.02	0.84	0.386 ± 0.148	0.82	0.371 ± 0.131	0.97	0.94
0.5	Sine	0.01	0.87	0.427 ± 0.130	0.83	0.367 ± 0.080	0.75	0.58

Note: The coverage and Schruben estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

in Figure 12, repeatedly traverses the steady-state mean. When the transient observations pass near the estimated steady-state mean, a premature truncation point is sometimes selected. The transients with a smaller shift in the mean, $\Delta\mu_y(\sigma_y)$, are more difficult to detect.

The effect of the truncation point on the ability to draw inferences can be seen by constructing confidence intervals. Transient observations in truncated sequences often bias the mean estimates, resulting in lower than nominal coverage rate for the confidence intervals. In contrast, excess truncation reduces the precision of the mean estimate and increases the confidence interval half width.

Table 57 shows effectiveness of the confidence intervals constructed with the truncated sequences. The column labeled "Fail" indicates the rate of times that the truncation sequence failed to obtain statistically independent batches with Fishman's algorithm. (See

Equation (6) on page 27.)

The coverage rates and average confidence interval half widths are excellent. The baseline for comparison are the sequences truncated at the 500th observations. These sequences with no transient data are labeled under transient type as "None". These single-run confidence interval techniques are based on approximations to account for the correlation in the output sequence. Although few of the truncated sequences result in nominal confidence interval coverage rates of 0.9, they are as close as the baseline case, which has a perfect truncation point. Even though the exponentially decaying sinusoid transients generally result in selected truncation points prior to steady state, these transients contain both high and low data values which apparently cancel each other's effect. The transients with a small step change to steady state resulted in the lowest coverage rates. Perhaps these transient values are too small to be detected but significant enough to bias the mean estimate. None of the cases result in excessive truncation, so the average confidence interval half widths are also comparable to the baseline case.

The pass rates for Schruben's initial bias test, discussed in Section 2.4.1, are also presented. The $\chi^2(\sigma_{\bar{y}})$, Equation (9), rates are generally near the nominal rate of 0.9. Schruben's tests with the estimated variance $F(\hat{\sigma}_{\bar{y}}^2)$, Equation (10), provide questionable indications since the mean estimate variance $\hat{\sigma}_{\bar{y}}^2$ is based on the biased estimates of \hat{Q}_d and \hat{R} (See page 167).

Table 58 shows the average number of batches and batch size selected by Fishman's algorithm, Equation (6). Little variation between the different transient types is apparent.

In contrast to monitoring the MMAE filter probabilities to determine a truncation point, the MMAE time-varying mean estimate $\hat{\mu}_y(t_n)$ can be tracked. For a test of this approach, the truncation point is selected when the MMAE mean estimate $\hat{\mu}_y(t_n)$ is within 1.0 of the steady-state mean estimate \bar{y} for 5 consecutive observations. The results of this strategy are shown in Table 56.

Table 58. Fishman's Number and Size of Batches ($\text{Prob}(\hat{\mu}_y(t_n)) > 0.9$)

Transient	Number of Batches		Batch Sizes	
	$\hat{\mu}_M$	$\hat{\sigma}_M$	$\hat{\mu}_B$	$\hat{\sigma}_B$
None	32.83	13.32	56.66	31.47
2.0 Step	30.73	12.06	51.25	26.50
2.0 Ramp	31.52	13.16	54.72	29.27
2.0 Exp	31.42	12.24	52.20	28.15
2.0 Sine	42.57	19.25	59.34	37.47
1.5 Step	31.13	12.18	51.75	27.65
1.5 Ramp	32.60	13.23	53.53	28.86
1.5 Exp	31.38	12.64	53.72	27.78
1.5 Sine	38.38	17.59	63.55	34.88
1.0 Step	28.77	12.31	67.21	35.71
1.0 Ramp	32.84	13.18	54.96	27.47
1.0 Exp	32.35	13.39	56.01	28.51
1.0 Sine	36.99	15.47	62.09	28.91
0.5 Step	34.74	16.35	69.90	35.17
0.5 Ramp	33.49	14.69	65.55	32.86
0.5 Exp	34.25	14.23	56.06	28.08
0.5 Sine	35.35	12.82	61.22	21.96

Table 59. AR(2) Truncation Point Errors ($\hat{n}_o - n_o$) ($\text{ABS}(\hat{\mu}_y(t_n) - \bar{y}) < 1$)

$\Delta\mu_y(\sigma_y)$	Type	Fail	Min	Max	Avg	St Dev	MSE
AR(2) Data with $\phi_1 = 0.6$ and $\phi_2 = 0.3$							
2.0	Step	0.00	-499	1232	108.0	147.4	33385.6
2.0	Ramp	0.00	-214	734	35.4	128.4	17741.0
2.0	Exp	0.00	-206	887	92.9	115.7	22025.9
2.0	Sine	0.00	-499	103	-438.4	108.2	203952.7
1.5	Step	0.00	-499	788	1.5	244.2	59654.7
1.5	Ramp	0.00	-323	1223	14.2	142.0	20359.6
1.5	Exp	0.00	-320	884	45.9	153.4	25628.1
1.5	Sine	0.00	-499	114	-451.7	89.1	211956.7
1.0	Step	0.00	-499	828	-279.3	265.0	148225.7
1.0	Ramp	0.00	-499	733	-73.8	153.3	28933.8
1.0	Exp	0.00	-330	768	-43.6	144.4	22741.9
1.0	Sine	0.00	-499	230	-456.4	79.5	214582.1
0.5	Step	0.00	-499	650	-462.9	93.4	223054.0
0.5	Ramp	0.00	-499	620	-357.0	170.0	156352.2
0.5	Exp	0.00	-393	914	-137.8	138.2	38100.2
0.5	Sine	0.00	-499	-81	-463.3	64.8	218834.9

Note: Negative Truncation points indicate transient observations are retained.

By comparing Tables 56 and 59, the truncation-point errors for both MMAE approaches are seen to be very similar. The similarity is not surprising since the MMAE probability approach tracked a probability while the MMAE mean approach tracked the filter means weighted by their probabilities. Therefore, both approaches are based on the MMAE filter probabilities. The truncation-point selection approach based on $\text{ABS}(\hat{\mu}_y(t_n) - \bar{y}) < 1.0$ appears to work slightly better with the transients with larger shifts, such as $\Delta\mu_y(\sigma_y)$ of 1.5 or 2.0. In contrast, the approach based on $\text{Prob}(\hat{\mu}_y(t_n) = \bar{y}) > 0.9$ has smaller truncation-point errors for the transients with a relatively smaller shift in their mean. However, any tradeoffs between the two approaches is very dependent upon the arbitrary selected criteria of 0.9 and 1.0.

As seen from comparing Table 57 and Table 60, the confidence interval coverage rates and half widths are very similar between the two truncation-point selection rules. Since there is minimal change in the number or size of batches, they are not shown. Because the two truncation-point selection approaches have such similar results, only one is used for further testing. The rule based on two mean estimates being within a small tolerance has more potential for determining a heuristic to select the tolerance than the rule based on a filter probability exceeding an arbitrary level. Therefore, the next section applies the truncation-point selection rule based on $\text{ABS}(\hat{\mu}_y(t_n) - \bar{y}) < 1$ to actual discrete-event simulation output.

5.4.2 M/M/1 Queue Waiting Time Data. Detecting the transient in the time-series test cases is considerably easier than with a single simulation output sequence. In some actual simulation outputs, rather than an initial bias consisting of observations that are statistical outliers of the steady-state output distributions, the initial transient consists of observations from a similar, but skewed distribution, which varies with time. Transient values can not be identified simply by being statistical outliers of an estimated steady-state output distribution. But the mean of a group of transient values may be biased compared to the mean of a group of steady-state observations.

In Section 2.4.3, Figures 2 through 4 depict the empirical transient for the simulation of

Table 60. AR(2) Truncation Point \hat{n}_o Evaluations ($ABS(\hat{\mu}_y - \bar{y}) < 1$)

Transient		NOBM			OBM		Bias Test	
$\Delta\mu_{\mathbf{y}}(\sigma_{\mathbf{y}})$	Type	Fail	Cov	Half Width	Cov	Half Width	$F(\hat{\sigma}_{\mathbf{y}}^2)$	$\chi^2(\sigma_{\mathbf{y}}^2)$
AR(2) Data with $\phi_1 = 0.6$ and $\phi_2 = 0.3$								
	None	0.02	0.85	0.392 ± 0.146	0.84	0.376 ± 0.124	0.98	0.93
2.0	Step	0.05	0.84	0.407 ± 0.162	0.83	0.394 ± 0.145	0.95	0.92
2.0	Ramp	0.02	0.87	0.398 ± 0.154	0.86	0.382 ± 0.134	0.99	0.95
2.0	Exp	0.03	0.83	0.404 ± 0.152	0.82	0.389 ± 0.131	0.97	0.95
2.0	Sine	0.02	0.92	0.455 ± 0.136	0.92	0.428 ± 0.092	0.61	0.44
1.5	Step	0.18	0.84	0.410 ± 0.169	0.82	0.395 ± 0.149	0.81	0.78
1.5	Ramp	0.03	0.83	0.398 ± 0.154	0.81	0.382 ± 0.134	0.96	0.94
1.5	Exp	0.02	0.84	0.405 ± 0.158	0.83	0.390 ± 0.140	0.95	0.93
1.5	Sine	0.02	0.91	0.450 ± 0.157	0.88	0.402 ± 0.085	0.63	0.43
1.0	Step	0.41	0.65	0.444 ± 0.173	0.62	0.422 ± 0.151	0.49	0.34
1.0	Ramp	0.05	0.82	0.387 ± 0.144	0.80	0.372 ± 0.124	0.92	0.89
1.0	Exp	0.03	0.85	0.392 ± 0.153	0.84	0.378 ± 0.137	0.97	0.94
1.0	Sine	0.02	0.92	0.446 ± 0.156	0.87	0.384 ± 0.084	0.69	0.51
0.5	Step	0.15	0.67	0.383 ± 0.130	0.65	0.366 ± 0.118	0.76	0.52
0.5	Ramp	0.12	0.77	0.388 ± 0.128	0.74	0.370 ± 0.111	0.84	0.65
0.5	Exp	0.03	0.83	0.381 ± 0.142	0.81	0.366 ± 0.125	0.96	0.93
0.5	Sine	0.01	0.89	0.430 ± 0.128	0.84	0.367 ± 0.075	0.73	0.56

Note: The coverage and Schruben estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 61. Correlation Technique Parameter Estimation Summary

	μ_y	\bar{y}		$\hat{\phi}_1$		$\hat{\phi}_2$		\hat{k}	
		$\hat{\mu}_{\bar{y}}$	$\hat{\sigma}_{\bar{y}}$	$\hat{\mu}_{\hat{\phi}_1}$	$\hat{\sigma}_{\hat{\phi}_1}$	$\hat{\mu}_{\hat{\phi}_2}$	$\hat{\sigma}_{\hat{\phi}_2}$	$\hat{\mu}_{\hat{k}}$	$\hat{\sigma}_{\hat{k}}$
$\rho = 0.95$	10.00	10.03	5.61	0.95	0.04	0.04	0.04	1.00	0.00

an $M/M/1$ queue with traffic intensity ρ of 0.95. The transient can not be detected visually from a single run. After applying Welch's moving window technique to the average of 30 simulation replications, shown in Figure 4, the average end of transient is seen to be about 1,000 observations.

The same truncation selection procedure as applied on the time-series data is applied to simulation output of $M/M/1$ with a traffic intensity ρ of 0.95. As Howard [37] reports, the same MMAE filter spacing does not work well with different models. The MMAE filter spacing is significantly increased for the $M/M/1$ output sequences. The filters are positioned based on the minimum, mean, and maximum of the last half of the output sequence:

$$\hat{\mu}_j = \bar{y} + 0.9(\bar{y} - \delta_j) \text{ where } \delta_j = y_{\min}, \bar{y}, \text{ and } y_{\max} \quad (67)$$

The simulation model inputs are such that the traffic intensity ρ is 0.95 and the theoretical steady-state waiting time is 10.0 time units. The parameter estimates for 2,000 simulation runs are shown in Table 61. Generally, the output sequence is represented approximately as a pure AR(1). The system model is nearly a pure AR(1) since $\hat{\phi}_2$ is near zero and the estimated measurement noise variance \hat{R} is zero ($\hat{k} = 1.0$). Originally, it was hoped that the measurement noise might represent the "lack of fit" for assuming an AR(2) dynamics equation. However, this example seems to indicate that, if the dynamics model inadequately represents the data's correlation structure, the parameter estimation increases the Kalman filter gain to one in order to disregard the dynamics model predictions completely. But a Kalman filter gain of one indicates no measurement noise is needed to model the data. Thus, the more "lack of fit" of the dynamics model to the real data there is, the

Table 62. Truncation Points \hat{n}_o

Type	Fail	Min	Max	Avg	St Dev
0.95	0.001	7	8070	1321.6	1234.1

Table 63. Truncation Point \hat{n}_o Evaluations

ρ	Fail	NOBM		OBM		Bias Test $F(\hat{\sigma}_y^2)$
		Cov	Half Width	Cov	Half Width	
0.9	0.32	0.56	2.761 ± 2.589	0.55	2.662 ± 2.486	0.54

less measurement noise variance is estimated.

The average selected truncation point, shown in Table 62, is 1,321.6. This compares reasonably well with a truncation point of 1,000 identified by Welch's technique.

As shown in Table 63, the Non-Overlapping Batch Means (NOBM) and Overlapping Batch Means (OBM) coverage rates with the truncated sequences are only 0.56 and 0.55 for 90 percent nominal confidence intervals. Two possible reasons for the low coverage are biased mean estimates or Fishman's algorithm, which selects batch size, induces a bias. A biased mean estimate is unlikely since 10,000 observations are used and the average truncation point of 1321.6 is greater than the truncation point selected by Welch's algorithm. Fishman's algorithm appears to have difficulty with this output data since for 32 percent of the sequences no batch size is determined. These confidence intervals are constructed with number of batches summarized in Table 64. The effects of Fishman's algorithm are investigated further.

Table 64. Fishman's Number and Size of Batches

	Number of Batches		Batch Sizes	
	$\hat{\mu}_M$	$\hat{\sigma}_M$	$\hat{\mu}_B$	$\hat{\sigma}_B$
$\rho = 0.9$	17.46	7.89	601.44	256.68

Table 65 shows the empirical mean of the output distribution for every 250th entity through the simulated queue. The mean of the 1,000th observation is very near the theoretical steady-state mean of 10.0, and the transient appears to be effectively complete by the 2,000th observation. Therefore, the average truncation point of 1321.6 retains some biased data, but the magnitude of the bias appears to have been small. From examining the variation in the data over the 2,000 runs, the average selected truncation point seems reasonable.

Since the analytical alternatives for truncation-point selection are unreliable heuristics [24, 105], one evaluation approach is to compare the Kalman filter results to the "best possible". The "best possible" is determined by increasing the truncation point at increments and calculating the confidence intervals on systems with analytic steady-state values.

Periodic truncation points at every 250th observation are tested, and the results are shown in Table 66 for 2,000 replications. The first two columns are the mean estimates and associated standard deviation for each of the truncated sequences. The column labeled $\hat{\mu}_{\hat{\mu}_y}$ is the mean of the observations in the truncated sequence, and $\hat{\sigma}_{\hat{\mu}_y}$ is the standard deviation over the 2,000 replications. These columns indicate the bias for not truncating any observations is very small, and that small bias is removed with truncating several hundred initial data values. The coverage rates, for any of these truncation sequences, are significantly below the nominal rate of 90 percent. This may be due to the highly correlated nature of the output sequences, the run length, or Fishman's algorithm.

Table 66 indicates that Fishman's algorithm, Equation (6), to determine the number of statistically independent batches may induce some bias. At each of the horizontal lines, the rate of sequences failing Fishman's test for independence significantly increases. A probable cause of these increases is that Fishman's algorithm only considers batch sizes in powers of two. Therefore, as more data is truncated, the truncated data sequence is shorter, and some of the sequences have insufficient data to test for batch independence with double the batch size. At the same place that the failure rate increases, the coverage rate significantly decreases. Possibly, truncated sequences with large and highly correlated values fail the test

Table 65. Average Periodic Output Values for 2,000 Replications ($\mu_y = 10.0$)

n	$\hat{\mu}_{y_n}$	$\hat{\sigma}_{y_n}$
1	0.00	0.00
250	6.17	5.80
500	7.67	7.39
750	8.60	8.46
1000	9.25	8.98
1250	9.73	9.66
1500	9.61	9.70
1750	9.73	10.00
2000	9.90	10.27
2250	10.08	10.43
2500	9.83	10.28
2750	10.00	10.30
3000	10.02	10.40
3250	10.11	10.55
3500	9.54	10.14
3750	10.00	10.44
4000	10.24	10.54
4250	9.93	10.16
4500	9.66	9.95
4750	9.61	10.09
5000	9.36	10.15
5250	9.60	10.42
5500	10.08	10.87
5750	10.04	10.70
6000	10.11	10.55
6250	9.98	10.71
6500	9.74	10.70
6750	9.91	10.88
7000	9.95	10.96
7250	10.20	11.20
7500	10.17	11.17
7750	10.38	11.00
8000	10.33	11.15
8250	10.41	11.22
8500	10.36	11.42
8750	10.50	10.93
9000	10.14	10.64
9250	9.85	10.33
9500	9.77	10.09
9750	9.60	9.83

Table 66. Periodic Truncation Points (Fishman's Number of Batches)

\hat{n}_0	$\hat{\mu}_{\hat{\mu}_y}$	$\hat{\sigma}_{\hat{\mu}_y}$	Fishman's Algorithm			NOBM		OLMB		Bias Test
			Fails	Batches	Size	Cov	HW	Cov	HW	
0	9.66	3.80	0.12	16.27	756.55	0.62	3.172	0.61	3.069	0.42
250	9.80	3.89	0.13	16.13	748.54	0.62	3.237	0.61	3.128	0.45
500	9.88	3.98	0.13	15.88	725.23	0.63	3.279	0.61	3.138	0.48
750	9.92	4.06	0.14	15.78	709.65	0.64	3.265	0.62	3.124	0.49
1000	9.95	4.14	0.47	19.84	486.42	0.51	2.221	0.49	2.159	0.50
1250	9.96	4.22	0.48	20.27	465.38	0.47	2.163	0.46	2.115	0.51
1500	9.97	4.30	0.45	19.21	479.60	0.49	2.267	0.47	2.194	0.52
1750	9.99	4.38	0.46	19.40	460.95	0.48	2.242	0.47	2.188	0.52
2000	9.99	4.45	0.42	18.43	475.01	0.50	2.379	0.49	2.282	0.53
2250	9.99	4.53	0.44	18.65	454.59	0.47	2.340	0.46	2.259	0.54
2500	9.99	4.61	0.40	17.74	467.98	0.51	2.465	0.49	2.367	0.55
2750	10.00	4.69	0.41	17.92	447.73	0.48	2.438	0.47	2.346	0.56
3000	9.99	4.77	0.37	16.97	462.66	0.52	2.563	0.51	2.446	0.57
3250	9.99	4.86	0.38	17.07	442.38	0.50	2.540	0.48	2.436	0.57
3500	9.99	4.95	0.34	16.18	454.98	0.53	2.685	0.51	2.557	0.58
3750	10.00	5.06	0.36	16.21	434.75	0.52	2.670	0.50	2.547	0.40
4000	9.99	5.16	0.32	15.21	452.37	0.54	2.831	0.52	2.675	0.38
4250	9.99	5.27	0.32	15.22	429.58	0.53	2.804	0.51	2.655	0.42
4500	9.99	5.38	0.28	14.37	444.77	0.54	2.974	0.52	2.807	0.46
4750	10.01	5.49	0.28	14.24	423.19	0.52	2.963	0.50	2.820	0.51
5000	10.03	5.61	0.25	13.55	435.26	0.53	3.168	0.51	3.016	0.56
5250	10.06	5.72	0.25	13.60	408.13	0.53	3.154	0.50	2.977	0.60
5500	10.06	5.84	0.66	18.59	252.00	0.24	1.823	0.23	1.769	0.65
5750	10.06	5.94	0.63	17.92	249.53	0.27	1.888	0.25	1.831	0.67
6000	10.06	6.06	0.60	16.88	249.92	0.28	1.987	0.26	1.909	0.68
6250	10.07	6.17	0.57	16.20	246.80	0.29	2.074	0.28	1.993	0.70
6500	10.07	6.29	0.55	15.24	246.86	0.31	2.168	0.30	2.075	0.73
6750	10.09	6.41	0.52	14.52	243.69	0.33	2.271	0.32	2.168	0.75
7000	10.10	6.55	0.49	13.72	241.42	0.35	2.392	0.33	2.276	0.77
7250	10.12	6.68	0.44	12.77	240.18	0.38	2.574	0.36	2.434	0.80
7500	10.13	6.84	0.40	11.77	238.75	0.40	2.768	0.38	2.614	0.83
7750	10.11	7.00	0.80	17.82	128.89	0.08	1.453	0.08	1.414	0.86
8000	10.08	7.16	0.75	16.61	126.06	0.12	1.566	0.11	1.506	0.88
8250	10.04	7.31	0.69	14.66	125.69	0.13	1.702	0.12	1.627	0.91
8500	9.98	7.42	0.61	12.86	125.01	0.17	1.927	0.16	1.831	0.95
8750	9.88	7.61	0.52	11.07	123.50	0.22	2.198	0.21	2.076	0.98
9000	9.76	7.85	0.84	15.83	64.24	0.02	1.212	0.02	1.161	0.09
9250	9.67	8.18	0.70	12.36	63.92	0.05	1.471	0.05	1.395	0.93
9500	9.60	8.56	0.88	15.46	32.51	0.01	0.869	0.01	0.838	0.98

Table 67. Truncation Points \hat{n}_o (10 Batches)

Type	Fail	Min	Max	Avg	St Dev
0.95	0.001	7	8070	1364.7	1260.4

Table 68. Truncation Point \hat{n}_o Evaluations (10 Batches)

ρ	NOBM			OBM		Bias Test $F(\hat{\sigma}_q^2)$
	Fail	Cov	Half Width	Cov	Half Width	
0.95	0.00	0.64	3.785 ± 3.870	0.63	3.685 ± 3.783	0.54

for independent batches more often. As seen in Figure 2 on page 38, simulation output of $M/M/1$ queue with $\rho = 0.95$ occasionally has very large and highly correlated subsequences. In contrast, sequences with generally low values, often with idle servers causing independent subsequences, may tend to pass Fishman's test for independent batches. Since low biased sequences may pass Fishman's test and be used to form confidence intervals at a higher frequency, the coverage rates are significantly less than the nominal rate.

To eliminate the effects of Fishman's algorithm determining the number of statistically independent batches, one thousand replications are tested with the number of batches fixed to ten. The truncation-point algorithm is applied to these 1,000 simulation runs. The selected truncation points, summarized in Table 67, are similar to the previous test. However, Table 68 shows that fixing the number of batches at ten improved the coverage rate. Therefore, Fishman's algorithm to select batch sizes does induce a bias and reduces the coverage rates.

The periodic results with a fixed batch size of ten, shown in Table 69, have a higher coverage rate at every periodic truncation point. However, no significant degradation in coverage rate is apparent when the initial transient is not eliminated. The $M/M/1$ with $\rho = 0.95$ queuing model was selected because of its extreme positive correlation. The high correlation induces a relatively longer transient than other simulation models. Perhaps the

Table 69. Periodic Truncation Points (Number of Batches Fixed at 10)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	Fails	Batches	Size	NOBM		OLMB		Schrub
0	9.70	3.87	0.00	10.00	1000.00	0.65	3.654	0.64	3.552	0.41
250	9.83	3.97	0.00	10.00	975.00	0.65	3.713	0.65	3.588	0.45
500	9.91	4.06	0.00	10.00	950.00	0.66	3.756	0.65	3.619	0.48
750	9.96	4.14	0.00	10.00	925.00	0.66	3.787	0.65	3.643	0.49
1000	9.99	4.23	0.00	10.00	900.00	0.65	3.804	0.64	3.664	0.50
1250	10.02	4.31	0.00	10.00	875.00	0.66	3.823	0.64	3.681	0.50
1500	10.03	4.39	0.00	10.00	850.00	0.65	3.859	0.64	3.695	0.51
1750	10.05	4.47	0.00	10.00	825.00	0.65	3.895	0.63	3.703	0.51
2000	10.06	4.55	0.00	10.00	800.00	0.64	3.890	0.62	3.707	0.53
2250	10.07	4.63	0.00	10.00	775.00	0.64	3.879	0.63	3.710	0.54
2500	10.07	4.73	0.00	10.00	750.00	0.64	3.882	0.62	3.714	0.55
2750	10.08	4.82	0.00	10.00	725.00	0.64	3.908	0.62	3.718	0.55
3000	10.08	4.91	0.00	10.00	700.00	0.64	3.911	0.62	3.725	0.55
3250	10.08	5.01	0.00	10.00	675.00	0.63	3.903	0.62	3.734	0.57
3500	10.10	5.11	0.00	10.00	650.00	0.63	3.916	0.61	3.740	0.58
3750	10.11	5.21	0.00	10.00	625.00	0.63	3.940	0.61	3.740	0.39
4000	10.11	5.31	0.00	10.00	600.00	0.62	3.932	0.60	3.740	0.39
4250	10.11	5.42	0.00	10.00	575.00	0.60	3.914	0.58	3.747	0.42
4500	10.11	5.54	0.00	10.00	550.00	0.59	3.934	0.57	3.753	0.45
4750	10.14	5.68	0.00	10.00	525.00	0.58	3.941	0.55	3.756	0.50
5000	10.18	5.82	0.00	10.00	500.00	0.56	3.934	0.54	3.757	0.56
5250	10.19	5.97	0.00	10.00	475.00	0.56	3.946	0.53	3.751	0.60
5500	10.20	6.12	0.00	10.00	450.00	0.55	3.939	0.52	3.746	0.65
5750	10.21	6.26	0.00	10.00	425.00	0.56	3.941	0.54	3.749	0.67
6000	10.22	6.40	0.00	10.00	400.00	0.55	3.938	0.54	3.746	0.69
6250	10.22	6.54	0.00	10.00	375.00	0.55	3.937	0.53	3.730	0.72
6500	10.20	6.67	0.00	10.00	350.00	0.54	3.904	0.52	3.706	0.74
6750	10.19	6.80	0.00	10.00	325.00	0.53	3.900	0.51	3.687	0.75
7000	10.18	6.94	0.00	10.00	300.00	0.51	3.871	0.49	3.664	0.77
7250	10.20	7.06	0.00	10.00	275.00	0.51	3.848	0.49	3.637	0.80
7500	10.22	7.18	0.00	10.00	250.00	0.50	3.820	0.48	3.594	0.82
7750	10.20	7.27	0.00	10.00	225.00	0.46	3.765	0.44	3.537	0.85
8000	10.16	7.33	0.00	10.00	200.00	0.44	3.693	0.42	3.467	0.88
8250	10.12	7.38	0.00	10.00	175.00	0.44	3.610	0.42	3.380	0.91
8500	10.08	7.44	0.00	10.00	150.00	0.43	3.504	0.41	3.278	0.95
8750	9.99	7.55	0.00	10.00	125.00	0.39	3.367	0.37	3.149	0.99
9000	9.88	7.75	0.00	10.00	100.00	0.36	3.167	0.34	2.940	0.09
9250	9.77	8.08	0.00	10.00	75.00	0.30	2.879	0.29	2.676	0.93
9500	9.76	8.48	0.00	10.00	50.00	0.24	2.540	0.23	2.357	0.98
9750	9.70	8.99	0.00	10.00	25.00	0.16	1.999	0.15	1.849	1.00

initial transient is insignificant when the single-run confidence interval techniques are applied to one long simulation run.

5.5 Summary

This chapter develops and tests a truncation-point selection algorithm for a single simulation output sequence. An AR(2) with measurement noise model is estimated using the correlation technique. Based on the estimated model, MMAE is applied with various mean estimates. The truncation point is selected by either monitoring the probability of the filter with the mean estimate or by tracking the distance between the MMAE mean estimate and the assumed steady-state mean estimate. The method performs well on time-series data with induced transients and on output of simulations of an $M/M/1$ queue.

For simulation practitioners, passing Fishman's test for statistically independent batches may not be sufficient to ensure reliable estimates. In fact, as shown in the example for simulations of an $M/M/1$ queue, the sequences which passed Fishman's test often are biased low. As Schmeiser [83] states, "the run must be long enough to calculate a valid confidence interval." Passing Fishman's test does not ensure that the run length is long enough.

The effective attainment of steady state is determined based on a single simulation sequence. This seems to contradict Law and Kelton's [53:550] supposition that generally one run is not sufficient to estimate the truncation point. However, the transient's effect on drawing inferences may not be significant when confidence interval techniques use one long simulation run. The output of a simulation of an $M/M/1$ queue with traffic intensity ρ of 0.95 was selected because the high correlation should induce a relative long transient. But the initial transient is not found significant in decreasing the coverage rate when a single long run is used. Schmeiser [83] states that "initial transients are often a major factor when using independent replications." Therefore, an approach for multiple replications, very similar to this technique, is proposed and tested in Chapter VI.

VI. Multiple Replications Steady-State Identification: Multiple Model Adaptive Estimation (MMAE) Approach

6.1 Introduction

Besides the confidence interval approaches based on one long output sequence, confidence intervals can be constructed using replications of the simulation model with different random number generator seeds. The confidence interval technique is called the method of replications. (See page 26 for a discussion.) The advantage of the method of replications is that means of the replications are independent, so classical statistic techniques can be applied. The disadvantage is that each run passes through the transient phase. Because of the increase in transient data, selection of an appropriate truncation point becomes more critical to prevent bias in the overall mean estimate.

This chapter describes the MMAE algorithm to select the truncation points for multiple replications of univariate sequences. Evaluation measures are discussed, and a Monte Carlo analysis with twelve simulation models is presented.

6.2 Methodology

The methodology applied in this approach is the same as the MMAE technique for a single run developed in Chapter V. The only change is that the single sequence is replaced with the sequence of averages from the M independent replications. The parameter estimation, discussed in the Section 5.2, is applied again. Since a steady-state model is needed, the last half of the averaged output sequence is assumed to have effectively attained steady state and is used to estimate the steady-state parameters. Schruben [87] also uses the last half of the output sequence for estimating a steady-state model. The MMAE algorithm developed in Section 5.3 is applied again. After the estimated truncation point \hat{n}_0 is selected, each of the M sequences is truncated. The steps of the truncation point selection algorithm follow:

- Step 1.** Select the simulation output length N and number of replications M . Run M replications of the simulation model until N outputs occur and store the sequences of outputs.
- Step 2.** Construct a sequence of length N of the averages of the M replications.
- Step 3.** Based on the last half of the averaged sequence, determine an estimate of the steady-state mean \bar{y} with Equation (63) and the autocorrelation coefficients with Equation (64). Based on Equations (62) and (65), estimate $\hat{\phi}_1$, $\hat{\phi}_2$, and \hat{k} . (Initial Monte Carlo results used $\hat{k} = 1.0$ as recommended in Chapter V, but better results are obtained by using \hat{k} .)
- Step 4.** Apply the estimated Kalman filter to the last half of the average sequence and estimate the residual variance with the mean squared residuals, as shown in Equation (66). Find the minimum and maximum observation in the last half of the sequence of averages. Initialize an MMAE bank of three filters spaced according to minimum, mean \bar{y} , and maximum, as shown in Equation (67). Calculate the Kalman filter gain \mathbf{K} with Equation (51). The initial state estimates $\hat{\mathbf{x}}(t_0)$ are zero vectors, and the initial MMAE filter probabilities $p_j(t_0)$ are $\frac{1}{3}$.
- Step 5.** Process the observations in the averaged sequence through the MMAE bank of filters. For each observation, propagate the estimated state vector in each filter using Equations (22) and (45). Update the state estimates with Equation (25) using the observation minus that filter's mean as the measurement. Calculate the MMAE filter probabilities $p_j(t_n)$ with Equation (36). Using the MMAE filter probabilities, calculate the weighted MMAE mean estimate $\hat{\mu}_y(t_n)$ by Equation (39) where $\hat{\mathbf{a}}(t_n) = \hat{\mu}_y(t_n)$. When $\text{ABS}[\hat{\mu}_y(t_n) - \bar{y}]$ is less than a small tolerance for a few consecutive observations, select the truncation point \hat{n}_0 as t_n . (In these tests, the small tolerance is 1.0 and 5 consecutive observations are used. A relative criterion, such as the absolute difference being less than a small percentage of the sample mean, is probably more appropriate in general applications.) Before iterating, reset any near-zero filter probabilities to a very small tolerance to allow for continue adaptation in the MMAE algorithm.

The selected truncation point \hat{n}_0 can be applied to the M output sequences used in the algorithm or the truncation point can be applied to future replications. Pilot runs are sometimes used to estimate the truncation point to prevent any chance of statistical bias. However, Law and Kelton suggest that, if the resulting truncated sequences are substantially larger than the truncated portion, using the same runs for truncation-point identification and confidence interval construction is "probably safe" [53:552]. The same runs are used for both selecting the truncation points and making the confidence intervals in the Monte Carlo analysis.

6.3 Measures of Effectiveness

The performance measures are different than single-run approaches because different techniques to draw inferences are applicable with multiple replications. In this study, three of the measures used by Kelton and Law [44] are applied. The three measures are the Point Estimator Bias (PEBIAS), the Mean Absolute Deviation (MAD), and the method of replication coverage rates with the corresponding confidence interval half widths. These measures are applicable to model outputs for which the steady-state mean can be calculated analytically.

The PEBIAS measures the error in the mean estimated with the truncated sequences and the theoretical mean. From each truncated sequence of length N' , the mean estimate $y_m(N')$ is the average of N' observations. The estimated mean $\hat{\mu}_y$ is the grand mean of the M truncated sequences $\hat{\mu}_y = \frac{1}{M} \sum_{m=1}^M y_m(N')$, and the point estimator bias is

$$\text{PEBIAS} = \hat{\mu}_y - \mu_y$$

The PEBIAS is presented as a percentage of the theoretical mean μ_y , $100 \frac{\hat{\mu}_y - \mu_y}{\mu_y}$.

The MAD is the absolute value of the error between the estimated and theoretical mean:

$$\text{MAD} = |\hat{\mu}_y - \mu_y|$$

Therefore, the MAD is the absolute value of the PEBIAS. With repeated applications, negative and positive values of PEBIAS can cancel each other. In contrast, the MAD values are all positive.

The output analysis objective is to make inference about the system. These inferences usually are stated through point estimates with associated confidence intervals. Therefore, the coverage rates for the confidence intervals and their associated half widths are important evaluation measures. The method of replications, Equation (3), is used to calculate a confidence interval for the mean of the truncated sequence. Along with these performance measures, Schruben's initial bias test, Equation (10), can be applied to the truncated sequence of averaged output. Since the same biased variance of the mean estimate as Chapter V on page 167 is applied, the Schruben's test statistic is questionable. Except for the addition of Schruben's initial bias test, these are the same performance measures that Kelton and Law [44] apply. The same simulation models also are used in an attempt to make some performance comparisons.

6.4 Monte Carlo Results

In this section, several analyses are conducted. Kelton and Law's algorithm [44] requires the analyst to specify nine parameters. Because of the difficulty in determining appropriate parameter values and implementing their algorithm, a Monte Carlo analysis comparable to their results is conducted. In this analysis, the scalar \hat{k} is set to 1.0 as suggested in the last chapter. In the next Monte Carlo analysis, the estimated value of \hat{k} is used and the simulation run length is fixed. These results include evaluations of periodic truncation points for sensitivity analysis. The final two sections present excitation on the effects of simulation run length and MMAE filter spacing.

Eleven simulation models are used to test the algorithm in the first Monte Carlo analysis. The first nine are queuing systems with a mean interarrival time of 1.0. Three simulations of $M/M/1$ queues with first-in-first-out (FIFO) queue discipline, empty and idle initial conditions, and infinite queue capacity are used. The traffic intensity ρ is varied

between 0.8, 0.9, and 0.95. The remainder of the queuing simulations have a traffic intensity ρ of 0.8. Two variants of the $M/M/1$ queue are changing the queue discipline to last-in-first-out (LIFO) or changing the initial conditions to an initial queue length L_0 of ten. Another queuing simulation is an $E_4/M/1$ where interarrival times are a four-stage Erlang random variable [77:109]. The number of servers is increased to two, an $M/M/2$ queue, and to four, an $M/M/4$ model. The final queuing simulation is three $M/M/1$ queues in tandem with an interarrival rate at each queue of 1.0 and traffic intensities ρ of 0.5, 0.7, and 0.9. (For queues in series with exponentially distributed interarrival times and service times, the interarrival rates are equal.) The queues in tandem are labeled $M/M/1/M/M/1/M/M/1$.

The last two models are discrete-event simulation models of computer systems considered by Law and Carson [51]. The time-sharing computer system has 50 terminals submitting jobs to a FIFO queue at the CPU. Each computer operator "thinks" for an exponentially distributed time with a mean of 25.0 and then submits a job requesting an exponentially distributed service time with mean of 0.8. The CPU processes jobs in increments of 0.1 time units with a fixed overhead for each session of 0.015. Unfinished jobs enter the CPU queue. Completed jobs return to the terminal and the operator "thinks" again. The output is the sequence of response times to the terminals. The time-sharing computer model's analytical steady-state values are calculated as shown by Adiri and Avi-Itzhak [3].

The second computer simulation has eight jobs in the system. CPU service times are exponentially distributed with a mean of 1.0 time units. After leaving the CPU, the jobs enter one of two peripheral queues with probability of 0.5 each. The mean service times at the peripherals are exponentially distributed with mean of 2.0 time units. When a job completed service at the peripheral, another job instantaneously arrives in the CPU queue. Initial conditions are one peripheral queue with six jobs, one job in the other peripheral queue, and one job in the CPU queue. The output response is the time for each job to be processed by the CPU and a peripheral device.

The theoretical steady-state output response times for these models are shown in Table 70 [44]. The Monte Carlo analysis which follows is divided into four sections. In the first

Table 70. Theoretical Steady-State Means

Model	μ_y
$M/M/1 \rho = 0.8$	3.200
$M/M/1 \rho = 0.9$	8.100
$M/M/1 \rho = 0.95$	18.050
$M/M/1 \rho = 0.8$ LIFO	3.200
$M/M/1 \rho = 0.8 L_0 = 10$	3.200
$E_4/M/1 \rho = 0.8$	1.814
$M/M/2 \rho = 0.8$	2.844
$M/M/4 \rho = 0.8$	2.386
$M/M/1/M/M/1/M/M/1$	10.233
Time-Sharing Computer	21.384
Central Server Computer	10.000

section, simulation runs are made under the same conditions as Kelton and Law [44]. The next section shows runs at a consistent simulation run length. Two sections of analytical excursions are included; one section examines the effect of increasing the simulation run lengths, and the other section examines the effect of the filter spacing.

6.4.1 Results with $\hat{k} = 1.0$ This Monte Carlo analysis is conducted in order to compare results with Kelton and Law [44]. As recommended in Chapter V, the MMAE approach is implemented with \hat{k} arbitrarily set equal to one.

Kelton and Law [44] propose a methodology to determine the appropriate simulation run length N and truncation point \hat{n}_0 for replicated simulations. Their algorithm is discussed on page 2.4.2. Their algorithm determines the run length by restarting all of the M simulations if the length is found to be insufficient. Restarting M replications is considered impractical for many discrete-event simulation applications. Because of the difficulty in setting the parameters and applying Kelton and Law's algorithm (See discussion beginning on page 2.4.2.), their algorithm is not implemented. However, in order to make comparisons on as equitable a basis as possible, the same simulation models and evaluation measures are used. In addition, the models are run for the same number of replications, five, and for the

Table 71. Correlation Technique Parameter Estimation Summary ($\hat{k} = 1.0$)

Data Model	μ_y	\bar{y}		$\hat{\phi}_1$		$\hat{\phi}_2$		\hat{k}	
		$\hat{\mu}_{\bar{y}}$	$\hat{\sigma}_{\bar{y}}$	$\hat{\mu}_{\hat{\phi}_1}$	$\hat{\sigma}_{\hat{\phi}_1}$	$\hat{\mu}_{\hat{\phi}_2}$	$\hat{\sigma}_{\hat{\phi}_2}$	$\hat{\mu}_{\hat{k}}$	$\hat{\sigma}_{\hat{k}}$
$M/M/1 \rho = 0.80$	3.20	3.21	0.53	0.88	0.06	0.07	0.05	1.00	0.00
$M/M/1 \rho = 0.90$	8.10	8.10	2.14	0.92	0.06	0.06	0.05	1.00	0.00
$M/M/1 \rho = 0.95$	18.05	17.30	5.89	0.92	0.05	0.06	0.05	1.00	0.00
$M/M/1 \rho = 0.80$ LIFO	3.20	3.30	0.63	0.45	0.15	0.22	0.16	0.78	0.17
$M/M/1 \rho = 0.80 L_0 = 10$	3.20	3.20	0.58	0.88	0.06	0.07	0.05	1.00	0.00
$E_4/M/1 \rho = 0.80$	1.81	1.82	0.28	0.86	0.07	0.07	0.06	1.00	0.00
$M/M/2 \rho = 0.80$	2.84	2.86	0.56	0.69	0.09	0.27	0.09	0.97	0.02
$M/M/4 \rho = 0.80$	2.39	2.40	0.55	0.64	0.15	0.34	0.15	0.89	0.05
$M/M/1/M/M/1/M/M/1$	10.23	10.19	2.22	0.89	0.07	0.09	0.07	0.99	0.01
Time-Sharing Computer	21.38	21.39	0.83	0.15	0.49	0.29	0.60	0.25	0.35
Central Server Computer	10.00	10.01	0.18	0.38	0.18	0.38	0.23	0.42	0.17

run length of the average run length from their results.

Another difference in the results is that their algorithm always estimates a truncation point. For their results, the maximum simulation length is set to 3,000 observations and the maximum deletion amount of one half. Since 1,500 appears to be beyond the transient for these output sequences, their algorithm defaults to a "good" truncation point. Since the MMAE approach seldom fails to estimate a truncation point, this difference does not impact the comparison significantly.

A Monte Carlo analysis is conducted with 1,000 sets of runs with 5 replications per set. The MMAE parameters are estimated using the last half of the averaged output sequence. Table 71 shows the average parameter estimates and their standard deviations. The average initial mean estimate $\hat{\mu}_{\bar{y}}$ is very near the theoretical steady-state mean value μ_y , except for the $M/M/1$ model with traffic intensity ρ of 0.95. The estimated AR(2) with measurement noise model often has significant first autoregressive coefficients $\hat{\phi}_1$ with small second autoregressive coefficients $\hat{\phi}_2$ and no measurement noise, $\hat{k} = 1.0$. In general, these models' outputs are adequately represented by a pure AR(1) model.

Based on the scalar estimate \hat{k} and the variance of the residuals for the last half of the

Table 72. Noise Variances Estimation Summary ($\hat{k} = 1.0$)

Data Model	\hat{Q}_d		\hat{R}	
	$\hat{\mu}_{\hat{Q}_d}$	$\hat{\sigma}_{\hat{Q}_d}$	$\hat{\mu}_{\hat{R}}$	$\hat{\sigma}_{\hat{R}}$
$M/M/1 \rho = 0.80$	0.23	0.02	0.00	0.00
$M/M/1 \rho = 0.90$	0.32	0.03	0.00	0.00
$M/M/1 \rho = 0.95$	0.39	0.06	0.00	0.00
$M/M/1 \rho = 0.80$ LIFO	13.57	10.32	4.14	6.12
$M/M/1 \rho = 0.80 L_0 = 10$	0.23	0.02	0.00	0.00
$E_4/M/1 \rho = 0.80$	0.13	0.01	0.00	0.00
$M/M/2 \rho = 0.80$	0.31	0.03	0.01	0.01
$M/M/4 \rho = 0.80$	0.38	0.07	0.05	0.02
$M/M/1/M/M/1/M/M/1$	0.58	0.04	0.01	0.00
Time-Sharing Computer	27.67	52.10	63.95	30.21
Central Server Computer	1.69	1.11	3.01	1.01

averaged output sequence, the dynamics noise variance \hat{Q}_d and the measurement noise variance \hat{R} are estimated. The estimation approach and its problems are discussed on pages 90 and 167. Table 72 shows the average estimates for the 1,000 Monte Carlo sets of runs. Whenever $\hat{k} = 1.0$, the measurement noise variance \hat{R} is also estimated to be zero.

Table 73 shows statistics on the selected truncation points \hat{n}_0 for the 1,000 Monte Carlo sets of runs. The column labeled 'Fail' indicates the proportion of sets for which the MMAE approach failed to select a truncation point. These sets of runs are excluded from further analysis. The averages of the MMAE selected truncation points are always less than $\frac{1}{2}$ of the average truncation points reported by Kelton and Law [44]. For the $E_4/M/1$ model, the average is only $\frac{1}{6}$ of the average they report. Kelton and Law report the accuracy for their estimation of the truncation point n_0 . Using this accuracy value to back out the variance, the MMAE truncation-point variance is less for every model except $M/M/4$. For the $M/M/4$, the truncation-point variances are almost equal. Therefore, the MMAE algorithm generally truncates less of the initial data than Kelton and Law's algorithm. If the MMAE truncation points are sufficient to remove the initial data bias, the resulting inferences are more precise because of the increased number of untruncated observations.

Table 73. Truncation Points \hat{n}_o ($\hat{k} = 1.0$)

Data Model	Run Length	Fail	Min	Max	Avg	St Dev
$M/M/1$ $\rho = 0.80$	1615	0.000	1	1422	178.3	184.9
$M/M/1$ $\rho = 0.90$	1910	0.013	11	1860	443.3	367.3
$M/M/1$ $\rho = 0.95$	2560	0.256	18	2424	667.1	458.7
$M/M/1$ $\rho = 0.80$ LIFO	1365	0.055	28	1360	335.3	286.6
$M/M/1$ $\rho = 0.80$ LIFO†	1365	0.045	24	1360	291.6	261.4
$M/M/1$ $\rho = 0.80$ $L_0 = 10$	1475	0.001	1	1373	121.0	154.5
$E_4/M/1$ $\rho = 0.80$	1460	0.000	1	1253	93.8	114.3
$M/M/2$ $\rho = 0.80$	1470	0.009	3	1434	308.1	309.2
$M/M/4$ $\rho = 0.80$	1390	0.489	8	1372	486.0	376.7
$M/M/1/M/M/1/M/M/1$	1860	0.004	23	1847	558.4	437.9
Time-Sharing Computer	1380	0.366	31	1375	125.3	142.9
Time-Sharing Computer †	1380	0.109	37	1324	122.6	166.9
Central Server Computer	1150	0.128	8	1110	89.4	148.9

† Truncation algorithm used the estimated \hat{k} rather than $\hat{k} = 1.0$.

The simulation models for which the average of \hat{k} , $\hat{\mu}_{\hat{k}}$, is less than one, shown in Table 71, also result in a high proportion of sets where MMAE failed to identify a truncation point. Because of the analysis of \hat{k} in Chapter V, the MMAE algorithm is implemented with \hat{k} set arbitrarily to one. For the two models with the lowest average for \hat{k} , $M/M/1$ LIFO and Time-Sharing Computer, the Monte Carlo analysis is run again with the MMAE algorithm modified to use the estimated \hat{k} rather than setting it to one. The results with the estimate \hat{k} , marked with the dagger †, show that this change reduces the rate of failures with only a small effect on the average selected truncation points. This change to using the estimated \hat{k} in the algorithm has little or no effect on most of the cases since their estimates of \hat{k} are close to 1.0.

Table 74 shows the measures of effectiveness for each of the tested models. The PEBIAS are larger than values reported by Kelton and Law [44] except for the $E_4/M/1$ and Central Server Computer models. Larger errors indicate worse results, but the impact is mitigated by two considerations. First, the percentages of PEBIAS are reported in terms of the steady-state means μ_v . Since these percentages are not large, the PEBIAS values are very small in

Table 74. Truncation Point \hat{n}_o Evaluations ($\hat{k} = 1.0$)

Data Model	PEBIAS (% of μ_y)	MAD	Cov	Half Width	Bias Test
$M/M/1$ $\rho = 0.80$	2.4 ± 0.79	0.3 ± 0.02	0.88 ± 0.017	0.8 ± 0.03	0.28
$M/M/1$ $\rho = 0.90$	2.8 ± 1.37	1.6 ± 0.07	0.82 ± 0.020	3.5 ± 0.15	0.54
$M/M/1$ $\rho = 0.95$	-14.2 ± 1.27	3.9 ± 0.14	0.69 ± 0.028	7.2 ± 0.27	0.55
$M/M/1$ $\rho = 0.80$ LIFO	8.2 ± 2.44	0.6 ± 0.07	0.88 ± 0.017	1.2 ± 0.12	0.39
$M/M/1$ $\rho = 0.80$ LIFO †	6.3 ± 2.11	0.5 ± 0.06	0.88 ± 0.017	1.1 ± 0.11	0.36
$M/M/1$ $\rho = 0.80$ $L_0 = 10$	1.8 ± 0.94	0.4 ± 0.02	0.88 ± 0.017	0.9 ± 0.04	0.21
$E_4/M/1$ $\rho = 0.80$	1.5 ± 0.78	0.2 ± 0.01	0.89 ± 0.016	0.4 ± 0.02	0.16
$M/M/2$ $\rho = 0.80$	6.4 ± 1.44	0.5 ± 0.03	0.88 ± 0.017	1.1 ± 0.06	0.56
$M/M/4$ $\rho = 0.80$	6.2 ± 2.34	0.5 ± 0.05	0.87 ± 0.024	1.1 ± 0.09	0.74
$M/M/1/M/M/1/M/M/1$	5.1 ± 1.33	1.9 ± 0.10	0.85 ± 0.018	4.1 ± 0.18	0.61
Time-Sharing Computer	-0.1 ± 0.22	0.5 ± 0.03	0.90 ± 0.020	1.3 ± 0.06	0.29
Time-Sharing Computer †	-0.1 ± 0.18	0.5 ± 0.03	0.89 ± 0.017	1.3 ± 0.03	0.39
Central Server Computer	0.1 ± 0.08	0.1 ± 0.01	0.90 ± 0.017	0.3 ± 0.01	0.28

† Truncation algorithm used the estimated \hat{k} rather than $\hat{k} = 1.0$.

absolute terms. In fact, the difference may be accounted for by the data variation between the two separate Monte Carlo analyses. The second consideration is that if the transient data biases the mean estimates, the PEBIAS has a sign indicative of the upward or downward bias induced by the initial conditions. Only two of the cases, the $M/M/1$ models with $\rho = 0.95$ and $L_0 = 10$, have the same sign of PEBIAS as is induced by the model's initial conditions. Apparently, the MMAE selected truncation points generally remove the bias induced by the initial conditions.

Whereas the PEBIAS allows positive and negative values to cancel, the absolute value in the MAD measure prevents errors from canceling each others effect. As a result, the MAD measure reflects both the accuracies and variances of the mean estimates. The MAD values are lower than the corresponding values reported by Kelton and Law [44] except for the $M/M/1$ LIFO and the $M/M/1/M/M/1/M/M/1$ models. For these two models, the reported MAD values are within 0.1 of their results. Therefore, in terms of MAD, the MMAE truncation points are generally better than the points selected by Kelton and Law's

technique.

The coverage rates for confidence intervals based on the sequences truncated at the MMAE selected truncation points are higher than reported by Kelton and Law [44] except for the $M/M/1$ with $\rho = 0.95$ model. In that case, the coverage rates are statistically indistinguishable, 0.69 and 0.70. Thus, in terms of coverage rates, the MMAE truncation points are superior. However, coverage rates should be examined in conjunction with their corresponding average confidence interval average half width. The average half widths are less except for three models. For the $M/M/1$ LIFO, $M/M/2$, and $M/M/4$ models, the average half widths are only 0.1 larger for the sequence truncated with the MMAE selected truncation points. The higher coverage with generally smaller average half width results from an increase in the number of steady-state observations since the MMAE selected truncation points are generally much earlier in the output sequence.

Overall, the MMAE selected truncation points appear superior to those selected by Kelton and Law's algorithm. The results are not conclusive since Kelton and Law's algorithm is not actually implemented. However, for this Monte Carlo analysis, which uses conditions as similar as possible to their Monte Carlo analysis, better results are obtained by the MMAE truncation-point selection algorithm.

6.4.2 Results with Estimated \hat{k} . As shown in Table 74, using the estimated \hat{k} generally improves the evaluations of the truncated sequence. Since better results are obtained when the estimated \hat{k} is used in the MMAE selection algorithm, the Monte Carlo analysis is conducted again. Besides using the estimate of \hat{k} , the simulation run length is fixed at 1,500 observations for each of the 5 replications.

One additional output sequence for the $M/M/1$ queue also is tested. The sequence consists of the length of the queue L_q sampled at intervals of one time unit. This output sequence for the model $M/M/1$ ($\rho = 0.8$ L_q) is an example of "statistics based on time-persistent variables" in contrast to all the other tested sequences, which are "statistics based on observations". (See discussion of output classifications on page 37.)

Table 75. Correlation Technique Parameter Estimation Summary (Estimated \hat{k})

Data Model	μ_y	\bar{y}		$\hat{\phi}_1$		$\hat{\phi}_2$		\hat{k}	
		$\hat{\mu}_{\bar{y}}$	$\hat{\sigma}_{\bar{y}}$	$\hat{\mu}_{\hat{\phi}_1}$	$\hat{\sigma}_{\hat{\phi}_1}$	$\hat{\mu}_{\hat{\phi}_2}$	$\hat{\sigma}_{\hat{\phi}_2}$	$\hat{\mu}_{\hat{k}}$	$\hat{\sigma}_{\hat{k}}$
$M/M/1 \rho = 0.80$	3.20	3.23	0.56	0.88	0.06	0.07	0.05	1.00	0.00
$M/M/1 \rho = 0.90$	8.10	8.15	2.33	0.91	0.06	0.07	0.06	1.00	0.00
$M/M/1 \rho = 0.95$	18.05	16.44	5.53	0.91	0.06	0.07	0.06	1.00	0.00
$M/M/1 \rho = 0.80$ LIFO	3.20	3.31	0.59	0.46	0.15	0.20	0.15	0.78	0.16
$M/M/1 \rho = 0.80 L_0 = 10$	3.20	3.20	0.56	0.88	0.06	0.07	0.05	1.00	0.00
$M/M/1 \rho = 0.80 L_q$	3.20	3.20	0.59	0.88	0.06	0.07	0.05	1.00	0.00
$E_4/M/1 \rho = 0.80$	1.81	1.82	0.27	0.86	0.07	0.07	0.05	1.00	0.00
$M/M/2 \rho = 0.80$	2.84	2.87	0.56	0.70	0.09	0.26	0.09	0.97	0.02
$M/M/4 \rho = 0.80$	2.39	2.41	0.54	0.64	0.15	0.34	0.15	0.89	0.05
$M/M/1/M/M/1/M/M/1$	10.23	10.25	2.54	0.88	0.08	0.10	0.07	0.99	0.01
Time-Sharing Computer	21.38	21.39	0.84	0.12	0.49	0.33	0.61	0.25	0.35
Central Server Computer	10.00	10.00	0.16	0.41	0.17	0.36	0.22	0.41	0.15

Table 75 shows a summary of the estimated system parameters. Except for the $M/M/1$ with $\rho = 0.95$ model, the average of the mean estimates are very close to the theoretical models. The $M/M/1$ with $\rho = 0.95$ results are discussed later in Section 6.4.3. Table 76 presents the corresponding estimates of the dynamics noise variance \hat{Q}_d and the measurement noise variance \hat{R} .

Table 77 shows the truncation-point statistics resulting with the MMAE algorithm using the estimated \hat{k} . Even when the run length is slightly longer than the Monte Carlo analysis with \hat{k} fixed at 1.0, the average truncation point is slightly earlier in the output sequence. The proportion of times that the MMAE algorithm failed to find a truncation point significantly decreases, except for the output from $M/M/1$ with $\rho = 0.95$. For this model, the failure rate increases because the run length is reduced from 2,560 observations to only 1,500 output values.

Table 78 shows the evaluation measures for the modified MMAE algorithm. The slight changes from the case in which the MMAE used $\hat{k} = 1.0$ are a result of different run lengths, different stochastic realizations, and slightly earlier truncation points. With the decrease in

Table 76. Noise Variances Estimation Summary (Estimated \hat{k})

Data Model	\hat{Q}_d		\hat{R}	
	$\hat{\mu}_{\hat{Q}_d}$	$\hat{\sigma}_{\hat{Q}_d}$	$\hat{\mu}_{\hat{R}}$	$\hat{\sigma}_{\hat{R}}$
$M/M/1 \rho = 0.80$	0.23	0.02	0.00	0.00
$M/M/1 \rho = 0.90$	0.32	0.04	0.00	0.00
$M/M/1 \rho = 0.95$	0.40	0.07	0.00	0.00
$M/M/1 \rho = 0.80$ LIFO	13.99	11.34	4.21	7.10
$M/M/1 \rho = 0.80 L_0 = 10$	0.23	0.02	0.00	0.00
$M/M/1 \rho = 0.80 L_q$	0.29	0.02	0.00	0.00
$E_4/M/1 \rho = 0.80$	0.13	0.01	0.00	0.00
$M/M/2 \rho = 0.80$	0.31	0.03	0.01	0.01
$M/M/4 \rho = 0.80$	0.37	0.07	0.05	0.02
$M/M/1/M/M/1/M/M/1$	0.58	0.05	0.01	0.00
Time-Sharing Computer	29.18	55.26	63.90	30.82
Central Server Computer	1.61	1.00	3.09	0.89

Table 77. Truncation Points \hat{n}_o (Estimated \hat{k})

Data Model	Run Length	Fail	Min	Max	Avg	St Dev
$M/M/1 \rho = 0.80$	1500	0.000	1	1317	179.9	199.5
$M/M/1 \rho = 0.90$	1500	0.017	9	1458	405.8	315.0
$M/M/1 \rho = 0.95$	1500	0.356	40	1440	512.8	328.7
$M/M/1 \rho = 0.80$ LIFO	1500	0.031	29	1456	309.9	284.6
$M/M/1 \rho = 0.80 L_0 = 10$	1500	0.001	1	1221	117.4	135.6
$M/M/1 \rho = 0.80 L_q$	1500	0.000	1	1489	207.1	221.3
$E_4/M/1 \rho = 0.80$	1500	0.000	1	1169	91.7	112.0
$M/M/2 \rho = 0.80$	1500	0.006	5	1482	300.3	304.3
$M/M/4 \rho = 0.80$	1500	0.372	14	1493	513.6	408.0
$M/M/1/M/M/1/M/M/1$	1500	0.008	16	1495	488.5	353.2
Time-Sharing Computer	1500	0.130	42	1332	118.8	149.0
Central Server Computer	1500	0.005	7	1446	38.8	95.6

Table 78. Truncation Point \hat{n}_o Evaluations (Estimated \hat{k})

Data Model	PEBIAS (% of μ_y)	MAD	Cov	Half Width	Bias Test
$M/M/1$ $\rho = 0.80$	3.0 ± 0.94	0.4 ± 0.02	0.86 ± 0.018	0.9 ± 0.03	0.30
$M/M/1$ $\rho = 0.90$	3.4 ± 1.52	1.7 ± 0.09	0.83 ± 0.020	3.9 ± 0.17	0.60
$M/M/1$ $\rho = 0.95$	-22.1 ± 1.48	5.1 ± 0.18	0.56 ± 0.032	7.3 ± 0.32	0.61
$M/M/1$ $\rho = 0.80$ LIFO	5.0 ± 1.30	0.4 ± 0.04	0.86 ± 0.018	1.0 ± 0.05	0.32
$M/M/1$ $\rho = 0.80$ $L_0 = 10$	1.1 ± 0.79	0.4 ± 0.02	0.87 ± 0.018	0.8 ± 0.03	0.21
$M/M/1$ $\rho = 0.80$ L_q	3.2 ± 1.03	0.4 ± 0.02	0.86 ± 0.018	1.0 ± 0.05	0.34
$E_4/M/1$ $\rho = 0.80$	1.0 ± 0.65	0.2 ± 0.01	0.89 ± 0.017	0.4 ± 0.01	0.15
$M/M/2$ $\rho = 0.80$	6.4 ± 1.32	0.4 ± 0.03	0.88 ± 0.017	1.0 ± 0.05	0.56
$M/M/4$ $\rho = 0.80$	8.3 ± 2.40	0.5 ± 0.05	0.86 ± 0.023	1.1 ± 0.08	0.76
$M/M/1/M/M/1/M/M/1$	4.6 ± 1.39	2.0 ± 0.10	0.84 ± 0.019	4.3 ± 0.19	0.66
Time-Sharing Computer	-0.2 ± 0.17	0.5 ± 0.02	0.90 ± 0.017	1.2 ± 0.03	0.36
Central Server Computer	0.1 ± 0.06	0.1 ± 0.00	0.90 ± 0.016	0.2 ± 0.00	0.19

run length, the $M/M/1$ with $\rho = 0.95$ coverage rate decreases from 0.69 to only 0.56. For this case, the effect of increasing the run length is examined in Section 6.4.3.

The following sequence of tables characterizes the transient and its effect for each of the eleven models. For each model, two tables are presented. The first table shows the average values of periodic output values and their variance. When the average approaches the steady-state value, the transient is essentially complete. These values are obtained by averaging over the 1,000 Monte Carlo sets each with 5 replications. Table 79 shows the results for the output of the $M/M/1$ queue simulation with $\rho = 0.8$. The initial conditions of an empty queue and idle server cause the first customer always to have a waiting time of zero. The downward initial bias appears to have a very small effect by the 200th observation since their average of 3.16 is almost equal to the theoretical average of 3.2. Certainly, the initial bias is inconsequential by the 400th observation. The double horizontal lines indicate between which periodic values the average truncation point occurs. For the $M/M/1$ with $\rho = 0.8$ output, the average truncation point is 179.9, which is between 100 and 200.

The second table on the output of each model shows the effect of truncating at periodic

Table 79. Periodic Output Values Points for $M/M/1$ with $\rho = 0.8$ ($\mu_y = 3.2$)

n	$\hat{\mu}_{yn}$	$\hat{\sigma}_{yn}$
1	0.00	0.00
100	2.99	3.61
200	3.16	3.93
300	3.17	3.95
400	3.25	3.93
500	3.19	3.93
600	3.23	4.01
700	3.20	4.05
800	3.21	3.87
900	3.19	4.04
1000	3.21	3.98
1100	3.22	4.01
1200	3.22	3.86
1300	3.21	3.89
1400	3.22	3.91
1500	3.19	3.93

points. Since no alternative analytical truncation-point technique is widely accepted, these tables investigate the sensitivity of the MMAE selected truncation points. In the table for each model, the first two columns are the mean estimate and its variance when the output sequences are truncated at that point. The next columns present the point estimator bias (PEBIAS) as a percentage of the theoretical steady-state mean and the average of the mean absolute deviation (MAD) for the truncated sequences. The coverage rate and average confidence interval half width are also presented for the truncated sequences. The final column presents the ratio of averaged sequences which passed Schruben's test for no initial bias. Again, the interval containing the average of the MMAE selected truncation points is indicated with double horizontal lines.

Table 80 shows the inferences that result for periodic truncations of the output from the $M/M/1$ model with $\rho = 0.8$. The mean estimate after deleting the first 100 observations has no significant downward bias from the initial conditions. Deleting more of the initial

Table 80. Periodic Truncation Points for $M/M/1$ with $\rho = 0.8$ ($\mu_y = 3.2$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	3.16	0.40	-1.2 ± 0.65	0.3 ± 0.01	0.86 ± 0.018	0.8 ± 0.02	0.00
100	3.21	0.42	0.5 ± 0.68	0.3 ± 0.01	0.87 ± 0.018	0.8 ± 0.02	0.18
200	3.22	0.44	0.6 ± 0.71	0.3 ± 0.01	0.87 ± 0.018	0.8 ± 0.02	0.35
300	3.22	0.46	0.7 ± 0.74	0.4 ± 0.02	0.87 ± 0.018	0.9 ± 0.03	0.50
400	3.23	0.47	0.8 ± 0.77	0.4 ± 0.02	0.86 ± 0.018	0.9 ± 0.03	0.63
500	3.23	0.49	0.9 ± 0.80	0.4 ± 0.02	0.88 ± 0.017	0.9 ± 0.03	0.74
600	3.23	0.52	0.8 ± 0.84	0.4 ± 0.02	0.88 ± 0.017	1.0 ± 0.03	0.85
700	3.23	0.54	0.9 ± 0.88	0.4 ± 0.02	0.88 ± 0.017	1.0 ± 0.03	0.92
800	3.22	0.58	0.7 ± 0.95	0.5 ± 0.02	0.86 ± 0.018	1.1 ± 0.03	0.95
900	3.22	0.61	0.6 ± 0.99	0.5 ± 0.02	0.85 ± 0.018	1.1 ± 0.04	0.97
1000	3.22	0.65	0.7 ± 1.06	0.5 ± 0.02	0.85 ± 0.019	1.2 ± 0.04	0.95
1100	3.21	0.70	0.5 ± 1.14	0.5 ± 0.02	0.86 ± 0.018	1.3 ± 0.04	0.90
1200	3.20	0.79	0.1 ± 1.29	0.6 ± 0.03	0.83 ± 0.020	1.4 ± 0.05	0.82
1300	3.19	0.95	-0.2 ± 1.54	0.7 ± 0.03	0.81 ± 0.020	1.7 ± 0.06	0.72
1400	3.22	1.23	0.5 ± 1.99	0.9 ± 0.04	0.81 ± 0.021	2.1 ± 0.08	0.92

observations increases the variance in the mean estimate. With no deletion, $\hat{n}_0 = 0$, the PEBIAS has a negative sign because of the downward bias induced by the initial conditions. However, after deleting only 100 observations, the PEBIAS no longer shows the effects of the initial conditions. The MAD statistic is a function of both the point estimator bias and its variance. The average value of the MAD statistic increases steadily as more data is deleted and the variance of the point estimator $\hat{\mu}_y$ increases. The coverage rate never attains its nominal value of 0.9. Besides a slight increase in the coverage rate, the initial transient has almost no effect. However, deleting more than half the data causes the coverage rate to decrease. The average confidence interval half widths steadily increase with less data. The final column reports the percentage of truncated averaged sequences which passed Schruben's test for no initial bias. The nominal rate of 0.9 is not achieved until 700 observations are deleted. Schruben's test appears overly sensitive to potential outliers. Schruben's test also results in odd fluctuations as the length of the truncated sequence becomes very short.

The periodic output values for the simulation of the $M/M/1$ queue with $\rho = 0.9$ are

Table 81. Periodic Output Values Points for $M/M/1$ with $\rho = 0.9$ ($\mu_y = 8.1$)

n	$\hat{\mu}_{y_n}$	$\hat{\sigma}_{y_n}$
1	0.00	0.00
100	5.63	5.81
200	6.90	7.18
300	7.40	7.90
400	7.76	8.27
500	7.83	8.54
600	8.03	8.78
700	7.96	8.83
800	8.16	9.01
900	8.16	9.26
1000	8.16	9.04
1100	8.11	9.15
1200	8.19	8.88
1300	8.12	8.95
1400	8.13	8.81
1500	8.05	8.84

shown in Table 81. The exponential or geometric decay of the transient value to the steady-state mean of 8.1 is apparent. The effects of the transient appear to be inconsequential by the 800th observations. From Table 82, the initial transient effects on the mean estimates are seen to be negligible after deleting the first 500 observations. After deleting 500 initial observations, the PEBIAS no longer has the negative sign induced by the start-up conditions and the coverage rate has attained its maximum value. Deleting further observations increases the variance of the mean estimators, which causes the coverage rate to decrease even with wider confidence intervals. The average MMAE selected truncation point of 405.8 appears very reasonable.

The periodic output from the simulation of the $M/M/1$ queue with $\rho = 0.95$, shown in Table 83, indicate that many of the output sequences have not completed their initial transient within 1,500 observations. The average of the 1,000 Monte Carlo sets, with 5 replications each, never attains the steady-state mean of 18.05. Therefore, the average of

the mean estimate based on the last half of the data \bar{y} , shown in Table 75, is significantly below the steady-state mean.

For the $M/M/1$ with $\rho = 0.95$, Table 84 shows that the average mean estimate also never attains the theoretical value. Regardless of the truncation point, a downward bias is indicated by the negative sign on the PEBIAS. The coverage is always below the nominal rate of 0.9, but increases as more initial data is truncated. Because many of these output sequences generally continue to increase, the MMAE truncation-point selection algorithm fails to pick a truncation point for 35.6 percent of the sets.

When the MMAE algorithm selected a truncation point, the coverage rate is only 0.56. This is lower than any of the fixed periodic truncation points, probably because of the manner in which the output varies. As shown in Figure 2 in Section 2.4.3, $M/M/1$ with $\rho = 0.95$ simulation output has long subsequences of thousands of observations with low average values and occasional subsequences of about one thousand observations with very

Table 82. Periodic Truncation Points for $M/M/1$ with $\rho = 0.9$ ($\mu_y = 8.1$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	7.61	1.64	-6.1 ± 1.05	1.4 ± 0.05	0.77 ± 0.022	2.9 ± 0.11	0.00
100	7.86	1.75	-2.9 ± 1.12	1.4 ± 0.06	0.81 ± 0.020	3.1 ± 0.11	0.22
200	7.98	1.84	-1.5 ± 1.19	1.4 ± 0.06	0.82 ± 0.020	3.2 ± 0.12	0.48
300	8.04	1.94	-0.7 ± 1.25	1.5 ± 0.07	0.83 ± 0.020	3.4 ± 0.13	0.69
400	8.09	2.02	-0.1 ± 1.30	1.5 ± 0.07	0.83 ± 0.020	3.5 ± 0.14	0.70
500	8.12	2.10	0.2 ± 1.35	1.6 ± 0.07	0.83 ± 0.020	3.7 ± 0.14	0.75
600	8.13	2.19	0.4 ± 1.41	1.6 ± 0.08	0.83 ± 0.020	3.9 ± 0.15	0.75
700	8.15	2.27	0.6 ± 1.46	1.7 ± 0.08	0.83 ± 0.020	4.0 ± 0.16	0.83
800	8.15	2.39	0.6 ± 1.53	1.8 ± 0.08	0.81 ± 0.020	4.2 ± 0.17	0.82
900	8.14	2.48	0.5 ± 1.59	1.9 ± 0.08	0.82 ± 0.020	4.3 ± 0.17	0.81
1000	8.13	2.60	0.4 ± 1.67	2.0 ± 0.09	0.81 ± 0.020	4.5 ± 0.18	0.78
1100	8.12	2.75	0.2 ± 1.77	2.1 ± 0.09	0.81 ± 0.021	4.8 ± 0.19	0.78
1200	8.10	2.96	0.0 ± 1.90	2.3 ± 0.10	0.79 ± 0.021	5.1 ± 0.19	0.82
1300	8.06	3.18	-0.5 ± 2.05	2.5 ± 0.10	0.78 ± 0.021	5.6 ± 0.20	0.94
1400	8.07	3.49	-0.4 ± 2.24	2.8 ± 0.11	0.78 ± 0.022	6.2 ± 0.21	0.99

Table 83. Periodic Output Values Points for $M/M/1$ with $\rho = 0.95$ ($\mu_y = 18.05$)

n	$\hat{\mu}_{y_n}$	$\hat{\sigma}_{y_n}$
1	0.00	0.00
100	7.61	7.09
200	10.29	9.48
300	11.86	11.10
400	12.93	12.24
500	13.73	13.20
600	14.50	14.00
700	14.94	14.52
800	15.55	15.26
900	15.89	15.85
1000	16.14	16.20
1100	16.47	16.56
1200	16.73	16.75
1300	16.90	16.98
1400	17.01	17.07
1500	17.01	17.28

Table 84. Periodic Truncation Points for $M/M/1$ with $\rho = 0.95$ ($\mu_y = 18.05$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	14.05	3.75	-22.1 ± 1.08	4.8 ± 0.14	0.58 ± 0.026	6.9 ± 0.23	0.00
100	14.69	4.00	-18.6 ± 1.15	4.5 ± 0.14	0.63 ± 0.025	7.4 ± 0.24	0.12
200	15.13	4.26	-16.2 ± 1.23	4.4 ± 0.14	0.67 ± 0.024	7.8 ± 0.26	0.32
300	15.46	4.51	-14.4 ± 1.30	4.3 ± 0.15	0.69 ± 0.024	8.3 ± 0.28	0.54
400	15.74	4.75	-12.8 ± 1.37	4.4 ± 0.15	0.70 ± 0.024	8.7 ± 0.30	0.47
500	15.98	4.98	-11.5 ± 1.44	4.5 ± 0.16	0.72 ± 0.023	9.2 ± 0.31	0.50
600	16.19	5.20	-10.3 ± 1.50	4.6 ± 0.16	0.73 ± 0.023	9.7 ± 0.33	0.52
700	16.36	5.41	-9.3 ± 1.56	4.7 ± 0.17	0.73 ± 0.023	10.1 ± 0.34	0.78
800	16.51	5.64	-8.5 ± 1.63	4.8 ± 0.17	0.74 ± 0.023	10.6 ± 0.36	0.79
900	16.63	5.84	-7.9 ± 1.68	4.9 ± 0.18	0.77 ± 0.022	11.0 ± 0.37	0.77
1000	16.75	6.03	-7.2 ± 1.74	5.0 ± 0.19	0.77 ± 0.022	11.4 ± 0.38	0.81
1100	16.83	6.23	-6.8 ± 1.79	5.1 ± 0.19	0.77 ± 0.022	11.9 ± 0.39	0.87
1200	16.91	6.48	-6.3 ± 1.87	5.3 ± 0.20	0.77 ± 0.022	12.4 ± 0.40	0.93
1300	16.93	6.78	-6.2 ± 1.96	5.5 ± 0.21	0.78 ± 0.022	13.0 ± 0.40	0.98
1400	16.97	7.07	-6.0 ± 2.04	5.8 ± 0.22	0.80 ± 0.021	13.8 ± 0.41	0.99

Table 85. Periodic Output Values Points for $M/M/1$ LIFO ($\mu_y = 3.2$)

n	$\hat{\mu}_{yn}$	$\hat{\sigma}_{yn}$
1	0.00	0.00
100	2.68	7.21
200	3.07	9.66
300	3.24	10.71
400	3.45	11.32
500	3.18	10.21
600	3.10	10.06
700	3.19	10.62
800	3.57	12.89
900	3.22	10.52
1000	2.98	9.69
1100	3.06	9.84
1200	3.22	11.20
1300	3.22	11.21
1400	3.29	10.77
1500	15.86	28.97

long waiting times when the system is congested. If none of the five replications happen to have become congested within 1,500 observations, the mean estimate based on the last half of the data sequence \bar{y} is significantly low. As a result, the MMAE mean estimate $\hat{\mu}_y(t_n)$ very likely comes within 1.0 of \bar{y} and a truncation point is selected. Under these conditions, the mean estimate and confidence interval are biased downward and fail to cover the steady-state mean. This phenomenon explains the lower coverage rate for the MMAE truncation sequences compared to the sequences truncated at fix intervals.

The $M/M/1$ LIFO average observations, shown in Table 85, indicate that the significant effects of the transient are negligible after 300th observations. From Table 86, the truncation point apparently can be selected earlier in the output sequence without adverse effects.

The output from the simulation of the $M/M/1$ with initial queue length L_0 of 10 is an example of an initially over-congested system. Except for the first waiting time, which is

still zero, the entities during the initial transient have waiting times generally longer than the steady-state mean. Table 87 shows that, by the 300th observation, the average waiting time has decreased to less than the steady-state value. Table 88 indicates that no or very little truncation achieves the highest coverage rate with the smallest average confidence interval half widths. The MMAE algorithm generally selects very early truncation points and achieves good results.

The length of queue L_q is an example of "statistics based on time-persistent variable." The output $M/M/1 \rho = 0.8 L_q$ is generated by sampling the queue length every time unit. Therefore, unlike for the other tested models, the state transition matrix Φ represents the change over a fixed time interval rather than the change between sequential observations. Table 89 indicates that the output is very near its steady-state mean by the 200th observation and the initial transient effectively is complete by the 400th observation. From Table 90, the effect of the initial conditions is insignificant on coverage after 100 observations. The

Table 86. Periodic Truncation Points for $M/M/1$ LIFO ($\mu_v = 3.2$)

\hat{n}_0	$\hat{\mu}_v$	$\hat{\sigma}_{\hat{\mu}_v}$	PEBIAS (% of μ_v)	MAD	Cov	HW	Bias Test
0	3.15	0.39	-1.4 ± 0.64	0.3 ± 0.01	0.85 ± 0.019	0.7 ± 0.02	0.00
100	3.24	0.42	1.2 ± 0.68	0.3 ± 0.01	0.88 ± 0.017	0.8 ± 0.02	0.11
200	3.26	0.43	1.8 ± 0.71	0.3 ± 0.01	0.88 ± 0.017	0.8 ± 0.02	0.23
300	3.27	0.46	2.1 ± 0.75	0.4 ± 0.02	0.88 ± 0.017	0.9 ± 0.03	0.35
400	3.27	0.49	2.3 ± 0.79	0.4 ± 0.02	0.87 ± 0.017	0.9 ± 0.03	0.46
500	3.28	0.51	2.5 ± 0.82	0.4 ± 0.02	0.88 ± 0.017	1.0 ± 0.03	0.59
600	3.29	0.53	2.7 ± 0.87	0.4 ± 0.02	0.88 ± 0.017	1.0 ± 0.03	0.70
700	3.30	0.57	3.0 ± 0.93	0.4 ± 0.02	0.88 ± 0.017	1.1 ± 0.04	0.76
800	3.31	0.62	3.4 ± 1.01	0.5 ± 0.02	0.87 ± 0.018	1.1 ± 0.04	0.84
900	3.32	0.67	3.8 ± 1.09	0.5 ± 0.02	0.87 ± 0.017	1.2 ± 0.04	0.88
1000	3.34	0.72	4.5 ± 1.17	0.5 ± 0.03	0.87 ± 0.017	1.3 ± 0.04	0.93
1100	3.38	0.79	5.6 ± 1.28	0.6 ± 0.03	0.87 ± 0.017	1.4 ± 0.05	0.96
1200	3.41	0.90	6.7 ± 1.47	0.7 ± 0.03	0.86 ± 0.018	1.6 ± 0.05	0.97
1300	3.49	1.13	9.1 ± 1.84	0.8 ± 0.04	0.87 ± 0.017	1.9 ± 0.07	0.99
1400	3.79	1.73	18.4 ± 2.82	1.2 ± 0.07	0.86 ± 0.018	2.7 ± 0.12	0.99

Table 87. Periodic Output Values Points for $M/M/1$ with $L_0 = 10$ ($\mu_y = 3.2$)

n	$\hat{\mu}_{y_n}$	$\hat{\sigma}_{y_n}$
1	0.00	0.00
100	3.36	4.18
200	3.28	4.07
300	3.15	3.97
400	3.24	3.86
500	3.26	3.97
600	3.18	3.88
700	3.23	3.86
800	3.18	3.90
900	3.19	4.02
1000	3.29	4.05
1100	3.25	4.01
1200	3.21	3.93
1300	3.22	3.93
1400	3.20	3.86
1500	3.09	3.77

Table 88. Periodic Truncation Points for $M/M/1$ with $L_0 = 10$ ($\mu_y = 3.2$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	3.29	0.42	2.9 ± 0.68	0.3 ± 0.01	0.88 ± 0.017	0.8 ± 0.02	0.00
100	3.21	0.43	0.4 ± 0.69	0.3 ± 0.01	0.88 ± 0.017	0.8 ± 0.02	0.21
200	3.20	0.44	0.2 ± 0.71	0.3 ± 0.01	0.87 ± 0.017	0.8 ± 0.02	0.36
300	3.20	0.45	0.1 ± 0.74	0.4 ± 0.01	0.87 ± 0.018	0.8 ± 0.02	0.50
400	3.21	0.47	0.2 ± 0.76	0.4 ± 0.02	0.87 ± 0.018	0.9 ± 0.02	0.65
500	3.20	0.49	0.1 ± 0.79	0.4 ± 0.02	0.87 ± 0.017	0.9 ± 0.03	0.77
600	3.20	0.51	0.0 ± 0.84	0.4 ± 0.02	0.87 ± 0.017	0.9 ± 0.03	0.83
700	3.20	0.54	0.1 ± 0.88	0.4 ± 0.02	0.87 ± 0.018	1.0 ± 0.03	0.92
800	3.20	0.58	0.0 ± 0.94	0.4 ± 0.02	0.86 ± 0.018	1.1 ± 0.03	0.96
900	3.20	0.61	0.0 ± 1.00	0.5 ± 0.02	0.86 ± 0.018	1.1 ± 0.04	0.97
1000	3.19	0.65	-0.2 ± 1.06	0.5 ± 0.02	0.86 ± 0.018	1.2 ± 0.04	0.96
1100	3.18	0.70	-0.7 ± 1.13	0.5 ± 0.02	0.84 ± 0.019	1.3 ± 0.04	0.91
1200	3.18	0.78	-0.6 ± 1.28	0.6 ± 0.03	0.83 ± 0.020	1.4 ± 0.05	0.82
1300	3.18	0.87	-0.7 ± 1.42	0.7 ± 0.03	0.81 ± 0.020	1.6 ± 0.06	0.71
1400	3.14	1.10	-1.7 ± 1.80	0.9 ± 0.04	0.79 ± 0.021	1.9 ± 0.07	0.94

average MMAE selected truncation point is 207.1. For this example of "statistics based on time-persistent variables", the MMAE selected truncation point is both effective and efficient at reducing the effects of the initial conditions.

For the $E_4/M/1$ model, the periodic average output values, Table 91, and the periodic truncation points, Table 92, show that the significant transient effects are relatively short-lived. As such, the average truncation point selected by the MMAE algorithm is only 91.7.

For the $M/M/2$ simulation output, the averages of periodic observations and the evaluation of periodic truncation points are shown in Tables 93 and 94, respectively. Since the average truncation point selected by the MMAE algorithm is 300.3, this case provides the opportunity to examine the effect of the variance in the MMAE truncation points compared to a fixed truncation point of 300. Since the evaluations of the MMAE truncation points use the same output sequences to select the truncation point and to draw inferences, a bias may be present. For the 1,000 sets of runs, the percentage of PEBIAS is 6.4 for the MMAE approach compared to 0.8 for the fixed truncation point of 300. However, 6.4 percent of μ_y is only 0.18 in absolute terms. This is rather small compared to the standard deviation of the mean estimates $\hat{\sigma}_{\hat{\mu}_y}$ of 0.45. The average MAD values are about the same. The sequences truncated with the MMAE selected points have a slightly larger half width and therefore a coverage rate a little closer to the nominal rate. The largest difference is in the rate of passing Schruben's test for initial bias. The sequences truncated at the fixed point pass 75 percent of the time, while only 56 percent of the sequences truncated at the MMAE selected point pass Schruben's test.

Tables 95 and 96 indicate that the output from the $M/M/4$ model has a relatively short transient. The sequences truncated at the MMAE selected truncation point achieve a coverage of 0.87, as high as any of the fixed truncation points. While the statistics and parameters for these sequences appear similar to the other models, the MMAE algorithm fails to select a truncation point in 37.2 percent of the Monte Carlo sets of runs. These failures are caused by the filter spacing, which is examined later in Section 6.4.4.

Table 89. Periodic Output Values Points for $M/M/1 L_q (\mu_y = 3.2)$

n	$\hat{\mu}_{yn}$	$\hat{\sigma}_{yn}$
1	0.47	0.82
100	2.87	3.80
200	3.15	4.12
300	3.17	4.32
400	3.21	4.28
500	3.23	4.29
600	3.30	4.43
700	3.23	4.42
800	3.23	4.39
900	3.17	4.18
1000	3.17	4.27
1100	3.22	4.29
1200	3.17	4.25
1300	3.19	4.32
1400	3.21	4.33
1500	3.09	4.22

Table 90. Periodic Truncation Points for $M/M/1 L_q (\mu_y = 3.2)$

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	3.15	0.41	-1.6 ± 0.67	0.3 ± 0.01	0.84 ± 0.019	0.8 ± 0.02	0.00
100	3.20	0.44	-0.1 ± 0.71	0.3 ± 0.01	0.86 ± 0.018	0.8 ± 0.02	0.17
200	3.21	0.45	0.3 ± 0.73	0.4 ± 0.01	0.86 ± 0.018	0.8 ± 0.02	0.35
300	3.21	0.47	0.4 ± 0.77	0.4 ± 0.02	0.85 ± 0.019	0.9 ± 0.02	0.49
400	3.21	0.50	0.4 ± 0.82	0.4 ± 0.02	0.85 ± 0.019	0.9 ± 0.03	0.64
500	3.22	0.52	0.5 ± 0.85	0.4 ± 0.02	0.86 ± 0.018	1.0 ± 0.03	0.77
600	3.21	0.54	0.3 ± 0.88	0.4 ± 0.02	0.85 ± 0.018	1.0 ± 0.03	0.85
700	3.20	0.57	0.1 ± 0.92	0.5 ± 0.02	0.86 ± 0.018	1.1 ± 0.03	0.90
800	3.19	0.61	-0.3 ± 0.99	0.5 ± 0.02	0.85 ± 0.019	1.1 ± 0.03	0.94
900	3.20	0.66	-0.1 ± 1.07	0.5 ± 0.02	0.84 ± 0.019	1.2 ± 0.04	0.96
1000	3.21	0.71	0.4 ± 1.15	0.5 ± 0.02	0.84 ± 0.019	1.3 ± 0.04	0.96
1100	3.23	0.80	0.8 ± 1.30	0.6 ± 0.03	0.83 ± 0.019	1.4 ± 0.05	0.92
1200	3.21	0.90	0.3 ± 1.46	0.7 ± 0.03	0.81 ± 0.020	1.6 ± 0.06	0.85
1300	3.22	1.07	0.5 ± 1.74	0.8 ± 0.04	0.81 ± 0.021	1.8 ± 0.07	0.74
1400	3.20	1.28	0.1 ± 2.08	1.0 ± 0.04	0.79 ± 0.021	2.2 ± 0.09	0.89

Table 91. Periodic Output Values Points for $E_4/M/1$ ($\mu_y = 1.814$)

n	$\hat{\mu}_{y_n}$	$\hat{\sigma}_{y_n}$
1	0.00	0.00
100	1.73	2.38
200	1.82	2.53
300	1.80	2.52
400	1.87	2.54
500	1.80	2.45
600	1.82	2.55
700	1.82	2.55
800	1.81	2.53
900	1.78	2.50
1000	1.81	2.47
1100	1.82	2.49
1200	1.85	2.51
1300	1.85	2.51
1400	1.82	2.50
1500	1.75	2.46

Table 92. Periodic Truncation Points for $E_4/M/1$ ($\mu_y = 1.814$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	1.80	0.20	-1.0 ± 0.57	0.2 ± 0.01	0.88 ± 0.017	0.4 ± 0.01	0.00
100	1.82	0.21	0.2 ± 0.59	0.2 ± 0.01	0.89 ± 0.017	0.4 ± 0.01	0.15
200	1.82	0.21	0.1 ± 0.61	0.2 ± 0.01	0.88 ± 0.017	0.4 ± 0.01	0.34
300	1.82	0.22	0.2 ± 0.64	0.2 ± 0.01	0.89 ± 0.016	0.4 ± 0.01	0.47
400	1.82	0.23	0.2 ± 0.66	0.2 ± 0.01	0.89 ± 0.016	0.4 ± 0.01	0.61
500	1.82	0.24	0.2 ± 0.69	0.2 ± 0.01	0.89 ± 0.016	0.5 ± 0.01	0.73
600	1.82	0.25	0.2 ± 0.72	0.2 ± 0.01	0.88 ± 0.017	0.5 ± 0.01	0.81
700	1.82	0.26	0.4 ± 0.76	0.2 ± 0.01	0.89 ± 0.016	0.5 ± 0.01	0.88
800	1.82	0.28	0.1 ± 0.81	0.2 ± 0.01	0.88 ± 0.017	0.5 ± 0.02	0.93
900	1.81	0.30	0.0 ± 0.87	0.2 ± 0.01	0.87 ± 0.017	0.6 ± 0.02	0.97
1000	1.82	0.33	0.2 ± 0.95	0.3 ± 0.01	0.87 ± 0.018	0.6 ± 0.02	0.98
1100	1.81	0.37	-0.1 ± 1.06	0.3 ± 0.01	0.86 ± 0.018	0.7 ± 0.02	0.97
1200	1.82	0.43	0.1 ± 1.22	0.3 ± 0.01	0.86 ± 0.018	0.8 ± 0.03	0.94
1300	1.81	0.51	-0.2 ± 1.45	0.4 ± 0.02	0.83 ± 0.019	0.9 ± 0.03	0.82
1400	1.81	0.67	-0.2 ± 1.91	0.5 ± 0.02	0.81 ± 0.021	1.1 ± 0.05	0.79

Table 93. Periodic Output Values Points for $M/M/2$ ($\mu_y = 2.844$)

n	$\hat{\mu}_{y_n}$	$\hat{\sigma}_{y_n}$
1	0.00	0.00
100	2.64	3.53
200	2.82	3.84
300	2.85	3.83
400	2.89	3.84
500	2.85	3.80
600	2.88	3.89
700	2.87	3.99
800	2.84	3.79
900	2.82	3.99
1000	2.86	3.85
1100	2.88	3.93
1200	2.88	3.78
1300	2.85	3.75
1400	2.87	3.83
1500	2.81	3.84

Table 94. Periodic Truncation Points for $M/M/2$ ($\mu_y = 2.844$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	2.81	0.39	-1.4 ± 0.72	0.3 ± 0.01	0.86 ± 0.018	0.7 ± 0.02	0.00
100	2.86	0.41	0.5 ± 0.76	0.3 ± 0.01	0.87 ± 0.018	0.8 ± 0.02	0.35
200	2.86	0.43	0.7 ± 0.79	0.3 ± 0.01	0.86 ± 0.018	0.8 ± 0.02	0.60
300	2.87	0.45	0.8 ± 0.82	0.3 ± 0.02	0.86 ± 0.018	0.8 ± 0.03	0.75
400	2.87	0.47	0.9 ± 0.86	0.4 ± 0.02	0.87 ± 0.018	0.9 ± 0.03	0.87
500	2.87	0.48	1.0 ± 0.89	0.4 ± 0.02	0.88 ± 0.017	0.9 ± 0.03	0.92
600	2.87	0.51	0.9 ± 0.93	0.4 ± 0.02	0.87 ± 0.017	1.0 ± 0.03	0.95
700	2.87	0.54	1.0 ± 0.98	0.4 ± 0.02	0.87 ± 0.017	1.0 ± 0.03	0.97
800	2.86	0.58	0.7 ± 1.05	0.4 ± 0.02	0.86 ± 0.018	1.0 ± 0.03	0.96
900	2.86	0.60	0.6 ± 1.10	0.5 ± 0.02	0.85 ± 0.019	1.1 ± 0.04	0.96
1000	2.87	0.64	0.8 ± 1.17	0.5 ± 0.02	0.84 ± 0.019	1.2 ± 0.04	0.92
1100	2.86	0.69	0.5 ± 1.26	0.5 ± 0.02	0.85 ± 0.019	1.3 ± 0.04	0.87
1200	2.85	0.78	0.1 ± 1.42	0.6 ± 0.03	0.83 ± 0.020	1.4 ± 0.05	0.81
1300	2.84	0.93	-0.3 ± 1.70	0.7 ± 0.03	0.81 ± 0.020	1.6 ± 0.06	0.75
1400	2.86	1.21	0.5 ± 2.21	0.9 ± 0.04	0.81 ± 0.021	2.0 ± 0.08	0.95

Table 95. Periodic Output Values Points for $M/M/4$ ($\mu_y = 2.386$)

n	$\hat{\mu}_{y_n}$	$\hat{\sigma}_{y_n}$
1	0.00	0.00
100	2.22	3.39
200	2.37	3.62
300	2.35	3.65
400	2.42	3.61
500	2.36	3.62
600	2.41	3.73
700	2.41	3.77
800	2.40	3.67
900	2.40	3.80
1000	2.37	3.71
1100	2.43	3.77
1200	2.43	3.61
1300	2.34	3.54
1400	2.42	3.67
1500	2.35	3.69

Table 96. Periodic Truncation Points for $M/M/4$ ($\mu_y = 2.386$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	2.34	0.38	-1.8 ± 0.83	0.3 ± 0.01	0.85 ± 0.019	0.7 ± 0.02	0.00
100	2.40	0.40	0.4 ± 0.88	0.3 ± 0.01	0.86 ± 0.018	0.8 ± 0.02	0.81
200	2.40	0.42	0.6 ± 0.91	0.3 ± 0.01	0.86 ± 0.018	0.8 ± 0.02	0.93
300	2.40	0.44	0.8 ± 0.95	0.3 ± 0.01	0.86 ± 0.018	0.8 ± 0.03	0.97
400	2.41	0.46	0.9 ± 0.99	0.4 ± 0.01	0.85 ± 0.018	0.9 ± 0.03	0.96
500	2.41	0.47	1.0 ± 1.03	0.4 ± 0.02	0.87 ± 0.017	0.9 ± 0.03	0.90
600	2.41	0.49	1.0 ± 1.08	0.4 ± 0.02	0.86 ± 0.018	0.9 ± 0.03	0.83
700	2.41	0.52	1.0 ± 1.14	0.4 ± 0.02	0.87 ± 0.018	1.0 ± 0.03	0.82
800	2.40	0.56	0.7 ± 1.22	0.4 ± 0.02	0.86 ± 0.018	1.0 ± 0.03	0.85
900	2.40	0.58	0.5 ± 1.27	0.4 ± 0.02	0.85 ± 0.019	1.1 ± 0.04	0.84
1000	2.40	0.62	0.6 ± 1.35	0.5 ± 0.02	0.85 ± 0.019	1.1 ± 0.04	0.81
1100	2.39	0.67	0.3 ± 1.45	0.5 ± 0.02	0.84 ± 0.019	1.2 ± 0.04	0.80
1200	2.38	0.75	-0.2 ± 1.64	0.6 ± 0.02	0.83 ± 0.020	1.3 ± 0.04	0.82
1300	2.37	0.90	-0.7 ± 1.96	0.7 ± 0.03	0.80 ± 0.021	1.5 ± 0.06	0.89
1400	2.39	1.17	0.1 ± 2.55	0.9 ± 0.04	0.78 ± 0.022	1.9 ± 0.08	0.97

For the $M/M/1/M/M/1/M/M/1$ model output, shown in Table 98, the significant effects of the initial condition are negligible by the 600th observation. For the purposes of drawing inferences, Table 97 indicates that a truncation point about 500 is best. The average truncation point selected by the MMAE algorithm is 488.5.

The last two of the twelve models, Time-Sharing Computer and Central Server Computer, are more similar to applied discrete-event simulation models than the previous simulation of queuing systems. The output summaries for the Time-Sharing Computer model are in Tables 99 and 100, and the Central Server Computer model summaries are in Tables 101 and 102. Both of these models have significant transient effects that are relatively short-lived, and the MMAE algorithm selects appropriate truncation points.

Overall in the Monte Carlo analysis, the MMAE algorithm selected truncation points that achieved coverage near or better than the best coverage achieved by a fixed truncation point. The best results in terms of coverage are achieved with the most realistic simulation models.

The three lowest coverage rates correspond to the three models with the highest average estimates for $\hat{\phi}_1$. Since increased correlation in the output sequence requires more observations to achieve the same accuracy in the estimate, this seems to indicate that the models with high autocorrelation at lag one should have been run longer. The next section examines the effect of increasing the run length.

6.4.3 The Effect of Simulation Run Length. A possible indication that long simulation run lengths are necessary is an estimate for $\hat{\phi}_1$ near one. Models with output with high autocorrelation (indicated by high estimates of $\hat{\phi}_1$) require more observations in each run to obtain reliable estimates. As shown in Table 103, the $\hat{\phi}_1$ appears to be inversely correlated with the coverage rate. Each of the twelve models in the Monte Carlo analysis are listed in order of decreasing values of the average of the first autocorrelation coefficient estimate $\hat{\phi}_1$ from Table 75. The next column shows the resulting average coverage rate from the truncated sequences, as given in Table 78. In the analysis, the nominal coverage rate is 0.9.

Table 97. Periodic Output Values Points for $M/M/1/M/M/1M/M/1$ ($\mu_y = 10.233$)

n	$\hat{\mu}_{y_n}$	$\hat{\sigma}_{y_n}$
1	0.00	0.00
100	7.62	6.24
200	8.98	7.46
300	9.60	8.31
400	9.88	8.90
500	10.00	8.97
600	10.26	9.13
700	10.27	9.30
800	10.34	9.46
900	10.46	9.57
1000	10.20	9.31
1100	10.27	9.50
1200	10.26	9.38
1300	10.27	9.34
1400	10.15	9.32

Table 98. Periodic Truncation Points for $M/M/1/M/M/1M/M/1$ ($\mu_y = 10.233$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	9.69	1.71	-5.3 ± 0.87	1.5 ± 0.05	0.77 ± 0.022	3.0 ± 0.10	0.00
100	9.98	1.83	-2.5 ± 0.93	1.5 ± 0.06	0.82 ± 0.020	3.2 ± 0.11	0.30
200	10.10	1.93	-1.3 ± 0.98	1.5 ± 0.06	0.82 ± 0.020	3.3 ± 0.12	0.58
300	10.17	2.04	-0.6 ± 1.04	1.6 ± 0.07	0.83 ± 0.019	3.5 ± 0.12	0.77
400	10.21	2.14	-0.2 ± 1.09	1.6 ± 0.07	0.84 ± 0.019	3.6 ± 0.13	0.79
500	10.24	2.25	0.1 ± 1.14	1.7 ± 0.08	0.84 ± 0.019	3.8 ± 0.14	0.83
600	10.26	2.36	0.2 ± 1.20	1.8 ± 0.08	0.83 ± 0.019	3.9 ± 0.14	0.82
700	10.26	2.47	0.2 ± 1.26	1.9 ± 0.08	0.82 ± 0.020	4.1 ± 0.15	0.86
800	10.25	2.60	0.2 ± 1.32	2.0 ± 0.09	0.81 ± 0.020	4.3 ± 0.16	0.82
900	10.23	2.72	0.0 ± 1.38	2.1 ± 0.09	0.81 ± 0.021	4.4 ± 0.17	0.80
1000	10.21	2.82	-0.2 ± 1.44	2.1 ± 0.10	0.81 ± 0.020	4.7 ± 0.18	0.77
1100	10.21	2.92	-0.2 ± 1.49	2.2 ± 0.10	0.82 ± 0.020	4.9 ± 0.19	0.75
1200	10.16	3.06	-0.7 ± 1.56	2.3 ± 0.10	0.81 ± 0.021	5.3 ± 0.20	0.81
1300	10.13	3.24	-1.1 ± 1.65	2.5 ± 0.11	0.80 ± 0.021	5.8 ± 0.21	0.93

Table 99. Periodic Output Values Points for Time-Sharing Computer Model ($\mu_y = 21.38$)

n	$\hat{\mu}_{yn}$	$\hat{\sigma}_{yn}$
1	0.66	0.68
100	19.21	17.07
200	21.19	20.49
300	21.50	21.12
400	21.67	20.77
500	21.14	20.84
600	21.06	20.80
700	21.15	19.83
800	20.99	20.34
900	21.57	21.34
1000	21.43	21.06
1100	21.69	21.54
1200	21.03	20.55
1300	20.64	20.27
1400	20.96	20.30

Table 100. Periodic Truncation Points for Time-Sharing Computer Model ($\mu_y = 21.38$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	20.73	0.58	-3.0 ± 0.14	0.7 ± 0.02	0.75 ± 0.022	1.2 ± 0.02	0.00
100	21.34	0.62	-0.2 ± 0.15	0.5 ± 0.02	0.91 ± 0.015	1.2 ± 0.02	0.55
200	21.38	0.65	0.0 ± 0.16	0.5 ± 0.02	0.91 ± 0.015	1.3 ± 0.02	0.70
300	21.38	0.68	0.0 ± 0.17	0.5 ± 0.02	0.90 ± 0.015	1.3 ± 0.02	0.75
400	21.38	0.71	0.0 ± 0.17	0.6 ± 0.02	0.91 ± 0.015	1.4 ± 0.03	0.78
500	21.38	0.75	0.0 ± 0.18	0.6 ± 0.02	0.90 ± 0.015	1.5 ± 0.03	0.80
600	21.39	0.78	0.0 ± 0.19	0.6 ± 0.03	0.90 ± 0.016	1.5 ± 0.03	0.83
700	21.39	0.82	0.0 ± 0.20	0.6 ± 0.03	0.89 ± 0.016	1.6 ± 0.03	0.84
800	21.39	0.86	0.0 ± 0.21	0.7 ± 0.03	0.90 ± 0.015	1.7 ± 0.03	0.81
900	21.38	0.91	0.0 ± 0.22	0.7 ± 0.03	0.90 ± 0.016	1.9 ± 0.03	0.64
1000	21.38	0.98	0.0 ± 0.24	0.8 ± 0.03	0.89 ± 0.016	2.0 ± 0.04	0.78
1100	21.40	1.10	0.1 ± 0.27	0.9 ± 0.03	0.91 ± 0.015	2.2 ± 0.04	0.99
1200	21.36	1.28	-0.1 ± 0.31	1.0 ± 0.04	0.90 ± 0.016	2.6 ± 0.05	1.00
1300	21.35	1.50	-0.2 ± 0.36	1.2 ± 0.05	0.89 ± 0.016	3.0 ± 0.06	0.99

Table 101. Periodic Output Values Points for Central Computer Model ($\mu_y = 10.0$)

n	$\hat{\mu}_{yn}$	$\hat{\sigma}_{yn}$
1	0.97	0.98
100	9.96	5.30
200	10.00	5.23
300	10.12	5.31
400	10.04	5.32
500	10.00	5.28
600	10.06	5.40
700	10.07	5.38
800	10.11	5.26
900	9.92	5.34
1000	10.10	5.34
1100	10.07	5.45
1200	10.03	5.41
1300	9.91	5.18
1400	10.01	5.25

Table 102. Periodic Truncation Points for Central Computer Model ($\mu_y = 10.0$)

\hat{n}_0	$\hat{\mu}_y$	$\hat{\sigma}_{\hat{\mu}_y}$	PEBIAS (% of μ_y)	MAD	Cov	HW	Bias Test
0	9.97	0.11	-0.3 ± 0.06	0.1 ± 0.00	0.87 ± 0.017	0.2 ± 0.00	0.00
100	10.00	0.11	0.0 ± 0.06	0.1 ± 0.00	0.90 ± 0.016	0.2 ± 0.00	0.45
200	10.00	0.12	0.0 ± 0.06	0.1 ± 0.00	0.90 ± 0.016	0.2 ± 0.00	0.62
300	10.00	0.13	0.0 ± 0.07	0.1 ± 0.00	0.89 ± 0.016	0.2 ± 0.00	0.77
400	10.00	0.13	0.0 ± 0.07	0.1 ± 0.00	0.89 ± 0.016	0.3 ± 0.00	0.85
500	10.00	0.14	0.0 ± 0.07	0.1 ± 0.00	0.89 ± 0.016	0.3 ± 0.00	0.90
600	10.00	0.15	0.0 ± 0.08	0.1 ± 0.00	0.90 ± 0.016	0.3 ± 0.01	0.95
700	10.00	0.15	0.0 ± 0.08	0.1 ± 0.00	0.90 ± 0.016	0.3 ± 0.01	0.96
800	10.00	0.17	0.0 ± 0.09	0.1 ± 0.01	0.88 ± 0.017	0.3 ± 0.01	0.96
900	10.00	0.18	0.0 ± 0.09	0.1 ± 0.01	0.89 ± 0.016	0.3 ± 0.01	0.94
1000	10.00	0.20	0.0 ± 0.10	0.2 ± 0.01	0.89 ± 0.016	0.4 ± 0.01	0.97
1100	10.00	0.21	0.0 ± 0.11	0.2 ± 0.01	0.89 ± 0.016	0.4 ± 0.01	1.00
1200	10.00	0.25	0.0 ± 0.13	0.2 ± 0.01	0.88 ± 0.017	0.5 ± 0.01	0.99
1300	10.00	0.30	0.0 ± 0.16	0.2 ± 0.01	0.90 ± 0.016	0.6 ± 0.01	0.99

Table 103. Autocorrelation and Coverage Relationship

Model	$\hat{\mu}_{\hat{\phi}_1}$	Cov
$M/M/1 \rho = 0.95$	0.91	0.56
$M/M/1 \rho = 0.9$	0.91	0.83
$M/M/1/M/M/1/M/M/1$	0.88	0.84
$M/M/1 \rho = 0.8$	0.88	0.86
$M/M/1 L_q$	0.88	0.86
$M/M/1 L_0 = 10$	0.88	0.87
$E_4/M/1$	0.86	0.89
$M/M/2$	0.70	0.88
$M/M/4$	0.64	0.86
$M/M/1$ LIFO	0.46	0.86
Central Server Computer	0.41	0.90
Time-Sharing Computer	0.12	0.90

Correlation coefficient equals -0.38 .

As the average estimated value of $\hat{\phi}_1$ increases, the realized coverage rate decreases. In fact, from the 1,000 Monte Carlo sets of runs, the averages of $\hat{\phi}_1$, $\hat{\mu}_{\hat{\phi}_1}$, and the coverage rates have a correlation coefficient of -0.38 . Large positive autocorrelation, apparent by $\hat{\phi}_1$ close to 1.0, indicate that each additional observation provides less information. Therefore, models with large positive autocorrelation in their output sequences must have more observations to achieve the same statistical precision.

Since the model with the highest average value for $\hat{\phi}_1$ and lowest coverage is $M/M/1$ with $\rho = 0.95$, the run length for that model is increased. Table 104 depicts the parameter estimates as the run length is increased. Longer run lengths result in less bias and less variance in the initial mean estimate \bar{y} . The other parameter estimates do not change significantly as the run length increased.

The MMAE algorithm selects truncation points for an increasing percentage of the trials as the run length increases. Table 105 shows that the average selected truncation point \hat{n}_0 moves slightly toward the beginning of the sequence as the run length increases. Table 106 reveals that all of the performance measures, except Schruben's initial bias test,

Table 104. $M/M/1 \rho = 0.95$ Parameter Estimation Summary ($\mu_v = 18.05$)

Run Length	μ_v	\bar{y}		$\hat{\phi}_1$		$\hat{\phi}_2$		\hat{k}	
		$\hat{\mu}_{\bar{y}}$	$\hat{\sigma}_{\bar{y}}$	$\hat{\mu}_{\hat{\phi}_1}$	$\hat{\sigma}_{\hat{\phi}_1}$	$\hat{\mu}_{\hat{\phi}_2}$	$\hat{\sigma}_{\hat{\phi}_2}$	$\hat{\mu}_{\hat{k}}$	$\hat{\sigma}_{\hat{k}}$
2,500	18.05	17.45	5.97	0.93	0.05	0.06	0.05	1.00	0.00
5,000	18.05	17.99	5.34	0.94	0.04	0.05	0.04	1.00	0.00
15,000	18.05	18.07	3.71	0.96	0.03	0.03	0.03	1.00	0.00
20,000	18.05	17.97	3.15	0.96	0.03	0.03	0.03	1.00	0.00
25,000	18.05	18.00	2.84	0.97	0.03	0.02	0.03	1.00	0.00
30,000	18.05	17.95	2.62	0.97	0.02	0.02	0.02	1.00	0.00
35,000	18.05	18.01	2.36	0.97	0.02	0.02	0.02	1.00	0.00

Table 105. Truncation Points \hat{n}_o

Run Length	Fail	Min	Max	Avg	St Dev
2,500	0.271	24	2391	682.6	493.4
5,000	0.151	35	3673	783.9	589.1
15,000	0.011	58	4932	707.2	559.0
20,000	0.001	61	5180	648.1	489.0
25,000	0.000	58	5463	634.8	486.5
30,000	0.000	38	3758	607.6	426.7
35,000	0.000	103	5341	590.3	427.7

Table 106. Truncation Point \hat{n}_o Evaluations

Run Length	PEBIAS (% of μ_y)	MAD	Cov	Half Width	Bias Test
2,500	-15.0 \pm 1.32	4.0 \pm 0.16	0.67 \pm 0.029	6.9 \pm 0.27	0.58
5,000	-6.6 \pm 1.13	3.1 \pm 0.13	0.76 \pm 0.024	6.5 \pm 0.23	0.28
15,000	-1.1 \pm 0.79	2.1 \pm 0.09	0.84 \pm 0.019	4.8 \pm 0.15	0.05
20,000	-0.9 \pm 0.69	1.9 \pm 0.08	0.84 \pm 0.019	4.2 \pm 0.13	0.03
25,000	-0.7 \pm 0.62	1.7 \pm 0.07	0.85 \pm 0.018	3.9 \pm 0.11	0.02
30,000	-0.7 \pm 0.55	1.5 \pm 0.06	0.86 \pm 0.018	3.5 \pm 0.09	0.01
35,000	-0.6 \pm 0.50	1.4 \pm 0.05	0.86 \pm 0.018	3.2 \pm 0.08	0.01

improved with increased run lengths. Perhaps the estimate of $\hat{\phi}_1$ may provide a basis for a "rule of thumb" to determine a reasonable simulation run length.

6.4.4 The Effect of Filter Spacing. Between using the estimate \hat{k} instead of $\hat{k} = 1.0$ and increasing the run length, the MMAE algorithm has a very low rate of failure to select a truncation point for all of the models except the $M/M/4$ queue. Tables 95 and 96 indicate that the $M/M/4$ transient is effectively complete by the 500th observation. The average value for $\hat{\phi}_1$ is 0.64, relatively low, so the run length of 1,500 observations is sufficient. Compared to the other simulation models, the $M/M/4$ output has less variance. Less variance results in the minimum, mean, and maximum output values being closer. Since these values are used to space the MMAE filters, tighter filter spacing results. If the filters are spaced too close together, the filter probabilities fluctuate between filters.

The lower filter is positioned at consecutively lower fixed values to test the effect of increasing the filter spacing. Only the lower filter is moved since the transient bias induced by the empty and idle initial conditions result in low output values. Table 107 shows that, as the lower filter is moved downward, the MMAE algorithm selects a truncation point on more trials. The average truncation point is earlier in the sequence as the filter spacing increases. Table 108 depicts the effects of these earlier truncation points on the performance measures. The PEBIAS only has the same sign as the transient values when the low filter is at -4.0. The MAD improves with each increase in spacing. The coverage appears to be

Table 107. $M/M/4$ Truncation Points \hat{n}_o

Low Filter Position	Fail	Min	Max	Avg	St Dev
$\bar{y} - 0.9(\bar{y} - y_{\min})$	0.372	14	1493	513.6	408.0
0	0.353	18	1493	524.9	409.0
-1	0.244	1	1491	499.2	386.6
-2	0.109	1	1485	367.1	391.5
-3	0.001	1	1474	113.7	274.0
-4	0.000	1	1058	11.5	88.6

Table 108. $M/M/4$ Truncation Point \hat{n}_o Evaluations

Low Filter Position	PEBIAS (% of μ_y)	MAD	Cov	Half Width	Bias Test
$\bar{y} - 0.9(\bar{y} - y_{\min})$	8.3 ± 2.40	0.5 ± 0.05	0.86 ± 0.023	1.1 ± 0.08	0.76
0	8.5 ± 2.41	0.5 ± 0.05	0.87 ± 0.022	1.1 ± 0.08	0.76
-1	7.8 ± 2.07	0.5 ± 0.04	0.86 ± 0.021	1.1 ± 0.07	0.77
-2	7.0 ± 1.72	0.5 ± 0.03	0.85 ± 0.020	1.0 ± 0.06	0.65
-3	2.7 ± 1.36	0.4 ± 0.03	0.84 ± 0.019	0.9 ± 0.04	0.17
-4	-1.0 ± 0.96	0.3 ± 0.02	0.85 ± 0.019	0.7 ± 0.03	0.02

decreasing, but not a statistically significant amount. The half widths are smaller because of the increase in the number of untruncated observations.

For the case of $M/M/4$, the increased filter spacing reduces the MMAE algorithm's rate of failure to select a truncation point. However, extreme filter spacing, which results in selecting truncation points very early in the sequence, may have an adverse effect on the coverage rate for the truncated sequences.

6.5 Summary

This chapter proposes an algorithm to select the truncation point to eliminate the bias induced by initial conditions for multiple replications of discrete-event simulations. The procedure is based on MMAE with the necessary parameters estimated using the last half of the output sequences. After estimating an assumed steady-state AR(2) with measurement noise model, three Kalman filters are initialized. The Kalman filters are spaced based on the minimum, mean, and maximum value of the averaged output values. By processing the average output values through the bank of Kalman filters, the truncation point is selected when the time-varying MMAE mean estimate is within a small tolerance of the mean estimate for five consecutive observations. The algorithm performs well with the steady-state model estimated with the last half of the sequences and the small tolerance on the mean set to 1.0.

This approach is tested on twelve models in a Monte Carlo analysis of 1,000 sets with each set using 5 replications. The selected truncation points result in truncated sequences that provided excellent estimates of the steady-state values. The method appears to work for both types of output sequences: "statistics based on observations" and "statistics based on time-persistent variables". In the first case, the state transition matrix used in the Kalman filter is the relationship between sequenced observations. For the periodic samples of time-persistent variables, the state transition matrix is the relationship between samples equally spaced in time. In both cases, the parameters are estimated in the same manner. The MMAE-selected truncation points perform slightly better than Kelton and Law's [44] truncation points.

This approach has several advantages over Kelton and Law's technique. First, the analyst has fewer parameters to specify. The decisions to make in applying the MMAE algorithm concern the filter spacing and the tolerances on the MMAE mean estimate to select the truncation point. The tested values for the MMAE algorithm, which are selected without attempting to optimize, appear to provide relatively robust results. In addition, these algorithm settings can easily be determined from the output sequences for the analyst. Second, as seen in Chapter V with the time series models, this approach works for non-monotonic convergence to the steady-state mean. Third, rather than having to restart the simulation models, such as Kelton and Law's technique, the analyst can specify the replication run length. However, if a sequential approach that continues the replication runs is desired, the MMAE approach can be modified easily.

VII. Multivariate Steady-State Identification

7.1 Introduction

This chapter extends the univariate initial data truncation methodology from Chapter VI to application for multivariate output sequences. A first-order vector autoregressive with measurement noise model is proposed to approximate the simulation output sequences. An estimation scheme based on the sample autocorrelation matrices is presented. Using the estimated model and parameters, Multiple Model Adaptive Estimation (MMAE) with a bank of Kalman filters is applied. The truncation point is selected when a vector norm of a mean estimate minus the MMAE time-varying mean estimate are within a small tolerance. The multivariate algorithm is summarized and tested.

7.2 VAR(1) with Measurement Noise Model

For univariate output sequences, Steudel and Wu [97] and Schriber and Andrews [85] report that periodic samples of $M/M/1$ queue length can be adequately represented with an AR(1) model. Charnes [12] finds that multivariate output sequences of queue lengths can be represented as a first-order vector autoregressive, VAR(1), model. In this application, the proposed model assumes the filter-design states have a VAR(1) formulation, but the states are observed with an independent additive white noise vector. Thus, multivariate simulation output is formulated as a VAR(1) with measurement noise model. The proposed model is linear with time-invariant system matrices and constant noise covariances.

The average of multivariate sequences with S responses is modeled as follows. Let $\mathbf{x}(t_n)$ represent the $S \times 1$ state vector, Φ be the $S \times S$ time-invariant state transition matrix, and $\mathbf{w}_d(t_n)$ be the $S \times 1$ white noise vector, which is jointly normally-distributed with a zero mean vector and an $S \times S$ covariance matrix of \mathbf{Q}_d . The first-order vector autoregressive model VAR(1) for the filter-design state vector is

$$\mathbf{x}(t_n) = \Phi \mathbf{x}(t_{n-1}) + \mathbf{w}_d(t_n)$$

In this application, the state vector is time-varying representations of each of the S responses corrected for that response's mean. For example, if the first simulation response is queue length, the first element of the state vector $\mathbf{x}_1(t_n)$ is the modeled queue length at time t_n minus the expected queue length.

This formulation is a VAR(1) with measurement noise model, rather than a pure VAR(1), because the states are observed only with the addition of a vector of measurement noise. Define the $S \times 1$ measurement vector $\mathbf{z}(t_n)$ as the filter-design state vector $\mathbf{x}(t_n)$ plus an $S \times 1$ measurement noise vector $\mathbf{v}(t_n)$. Thus, the measurement matrix \mathbf{H} is simply an $S \times S$ identity matrix. The associated measurement model is

$$\begin{aligned}\mathbf{z}(t_n) &= \mathbf{H}\mathbf{x}(t_n) + \mathbf{v}(t_n) \\ &= \mathbf{x}(t_n) + \mathbf{v}(t_n)\end{aligned}$$

The simulation sequence averaged over the replications $\{\bar{\mathbf{y}}_n\}$ is used to determine realizations of the measurements:

$$\mathbf{z}_n = \bar{\mathbf{y}}_n - \hat{\boldsymbol{\mu}}_y \quad (68)$$

While significant lag effects in many simulation models may occur at greater than one lag, Charnes [12] reports that, for periodic samples of queue lengths, a VAR(1) formulation is sufficient to characterize the autocorrelation structure. The increased flexibility of the VAR(1) with measurement noise model should be sufficient for similar types of output sequences. However, analysts should carefully consider the applicability of this methodology to untested types of simulation models. Specifically, if each of the response sequences does not exhibit significant autocovariances at one and two lags, the inverse matrices necessary for estimation may not exist.

7.3 Parameter Estimation

In similar fashion as the univariate case, parameter estimates are based on the sample autocorrelations. Assuming covariance stationarity, the autocovariance matrices at a lag of i

for the sequence of $S \times 1$ averaged output vectors $\{y(t_1), y(t_2), \dots, y(t_N)\}$ are defined as

$$\Gamma_{yy}(i) \equiv E[(y(t_n) - \mu_y)(y(t_{n-i}) - \mu_y)^T]$$

The theoretical covariance matrix and first two lagged autocovariance matrices of $y(t_n)$ are derived below. Because the dynamics noise sequence $\{w_d(t_n)\}$ and the measurement noise sequence $\{v(t_n)\}$ are each modeled as zero mean, independent in time, and independent of each other and the state vectors, many of the cross terms have zero expected value:

$$\begin{aligned} \Sigma_{yy} &= \Gamma_{yy}(0) \\ &= E[(y(t_n) - \mu_y)(y(t_n) - \mu_y)^T] \\ &= E[(x(t_n) + v(t_n))(x(t_n) + v(t_n))^T] \\ &= E[x(t_n)x^T(t_n)] + E[v(t_n)x^T(t_n)] + E[v(t_n)x^T(t_n)] + E[v(t_n)v^T(t_n)] \\ &= E[x(t_n)x^T(t_n)] + E[v(t_n)v^T(t_n)] \\ &= \Gamma_{xx}(0) + R \\ &= P^- + R \end{aligned}$$

Since the measurement matrix H equals the identity matrix, the covariance of output sequence is equal to the Kalman filter predicted residual variance, $HP^-H^T + R$. The autoco-

variance matrix at one lag $\Gamma_{yy}(1)$ is

$$\begin{aligned}
\Gamma_{yy}(1) &= E[(y(t_n) - \mu_y)(y(t_{n-1}) - \mu_y)^T] \\
&= E[(x(t_n) + v(t_n))(x(t_{n-1}) + v(t_{n-1}))^T] \\
&= E[x(t_n)x^T(t_{n-1})] + E[v(t_n)x^T(t_{n-1})] + E[v(t_n)x^T(t_{n-1})] + E[v(t_n)v^T(t_{n-1})] \\
&= E[x(t_n)x^T(t_{n-1})] \\
&= \Gamma_{xx}(1) \\
&= E[(\Phi x(t_{n-1}) + w_d(t_n))x^T(t_{n-1})] \\
&= \Phi E[x(t_{n-1})x^T(t_{n-1})] \\
&= \Phi \Gamma_{xx}(0) \\
&= \Phi P^-
\end{aligned}$$

Similarly, the autocovariance matrix at lag two are derived:

$$\begin{aligned}
\Gamma_{yy}(2) &= E[(y(t_n) - \mu_y)(y(t_{n-2}) - \mu_y)^T] \\
&= E[(x(t_n) + v(t_n))(x(t_{n-2}) + v(t_{n-2}))^T] \\
&= E[x(t_n)x^T(t_{n-2})] \\
&= \Gamma_{xx}(2) \\
&= E[(\Phi x(t_{n-1}) + w_d(t_n))x^T(t_{n-2})] \\
&= \Phi E[x(t_{n-1})x^T(t_{n-2})] \\
&= \Phi \Gamma_{xx}(1) \\
&= \Phi^2 \Gamma_{xx}(0) \\
&= \Phi^2 P^-
\end{aligned}$$

Using these relationships, the state transition matrix Φ and the Kalman filter gain matrix K are determined in terms of the autocovariance matrices:

$$\begin{aligned}
\Gamma_{yy}(2)\Gamma_{yy}^{-1}(1) &= \Phi^2 P^- (\Phi P^-)^{-1} \\
&= \Phi^2 P^- (P^-)^{-1} \Phi^{-1} \\
&= \Phi
\end{aligned} \tag{69}$$

From Equation (30) on page 58, the Kalman filter gain \mathbf{K} are derived as follows:

$$\begin{aligned}
 \mathbf{K} &= \mathbf{P}^{-1} \mathbf{H}^T [\mathbf{H} \mathbf{P}^{-1} \mathbf{H}^T + \mathbf{R}]^{-1} \\
 &= \mathbf{P}^{-1} [\mathbf{P}^{-1} + \mathbf{R}]^{-1} \\
 &= \Phi^{-1} \Gamma_{yy}(1) \Gamma_{yy}^{-1}(0) \\
 &= [\Gamma_{yy}(2) \Gamma_{yy}^{-1}(1)]^{-1} \Gamma_{yy}(1) \Gamma_{yy}^{-1}(0) \\
 &= \Gamma_{yy}(1) \Gamma_{yy}^{-1}(2) \Gamma_{yy}(1) \Gamma_{yy}^{-1}(0)
 \end{aligned} \tag{70}$$

The inverse matrices necessary for estimation ($\Gamma_{yy}^{-1}(i)$ for $i = 0, 1, 2$) very likely exist if each of the output sequences has significant values for its first and second autocovariances. If each of these sequence exhibits AR(1) behavior, the first two autocovariances probably are not zero.

Using Equations (69) and (70) and the estimated autocovariance matrices based on the last half of the output sequence,

$$\hat{\Gamma}_{yy}(i) = \frac{2}{N-2} \sum_{n=\frac{N}{2}+1}^N (y_n - \bar{y})(y_{n-i} - \bar{y})^T \text{ for } i = 0, 1, 2 \tag{71}$$

From these sample autocovariance matrices, the state transition matrix and the Kalman filter gain vector can be estimated. These are the matrices necessary to apply the Kalman filter. Applying the Kalman filter to the last half of the sequence can determine the filter-design residual covariance matrix:

$$\mathbf{H} \mathbf{P}^{-1} \mathbf{H}^T + \mathbf{R} = \frac{2}{N-2} \sum_{n=\frac{N}{2}+1}^N \mathbf{r}_n \mathbf{r}_n^T \tag{72}$$

Using the last half of the output sequence, an assumed steady-state mean \bar{y} is calculated and the minimum and maximum of each response are found. All the estimates necessary to apply MMAE (Φ , \mathbf{K} , $\mathbf{H} \mathbf{P}^{-1} \mathbf{H}^T + \mathbf{R}$, and \bar{y}) are available. After spacing and initializing the bank of filters, MMAE can be applied. The $L = 3^S$ filters are positioned at each combination of the minimum, mean, and maximum of the S responses. The initial filter state vectors

are zeros. As in the univariate case, the MMAE time-varying mean estimate $\hat{\boldsymbol{\mu}}_{\mathbf{y}}(t_n)$ is compared with $\bar{\mathbf{y}}$ with a vector norm to determine the appropriate truncation point. When the Euclidean distance of $\|\hat{\boldsymbol{\mu}}_{\mathbf{y}}(t_n) - \bar{\mathbf{y}}\|$ is less than small tolerance for a few consecutive observations, the truncation point is selected. The small tolerance should probably be a small percent, perhaps ten, of the vector norm of the sample mean, but in these tests, an absolute value of one is used. Achieving the criterion for several consecutive observations reduces the chance that the randomness of the output sequence might give a false indication that its transient is effectively complete. The next section summarizes the specific steps in the technique.

7.4 *The Multivariate Truncation-Point Selection Algorithm*

The truncation-point selection algorithm for multivariate output sequences is summarized below.

- Step 1.** Select the simulation run length N and number of replications M . Run M replications of the simulation of length N , storing the multivariate sequences each with S responses.
- Step 2.** Construct a sequence of length N of the average across the M replications for each of the S responses.
- Step 3.** Based on the last half of the averaged sequence, determine an estimate of the steady-state mean vector $\bar{\mathbf{y}}$, the covariance matrix and autocovariance matrices with Equation (71). Based on Equations (69) and (70), estimate $\hat{\Phi}$ and \mathbf{K} . Estimate $\mathbf{HP} - \mathbf{H}^T + \mathbf{R}$ with Equation (72).
- Step 4.** Find the minimum and maximum of each of the S responses in the last half of the sequence of averages. Initialize an MMAE bank of $L = 3^S$ filters with a filter at each combination of the minimums, means, and maximums of the S responses. The initial state estimates $\hat{\mathbf{x}}(t_0)$ are zero vectors, and the initial MMAE filter probabilities $p_l(t_0)$ are $\frac{1}{L}$.

Step 5. Process the observations in the averaged sequence through the MMAE bank of filters. For each observation, propagate the estimated state vector in each filter using Equations (22) and (45). Update the state estimates with Equation (25), using the observation from the averaged sequence, \bar{y}_n , minus that filter's mean $\hat{\mu}_y^l$ as the measurement z_n . Calculate the MMAE filter probabilities $p_j(t_n)$ with Equation (36). Using the MMAE filter probabilities, calculate the weighted MMAE mean estimate $\hat{\mu}_y(t_n)$ by Equation (39) where $\hat{a}(t_n) = \hat{\mu}_y(t_n)$. Reset any near zero MMAE filter probabilities to a very small positive value, such as $\frac{0.05}{L}$. When $\|\hat{\mu}_y(t_n) - \bar{y}\| \leq 1.0$ for 5 consecutive observations, select the truncation point \hat{n}_0 as t_n . (In these tests, the Euclidean norm is used, the small tolerance is 1.0 and 5 consecutive observations are used. A relative criterion, such as the absolute difference being less than a small percentage of the vector norm of the sample mean, is probably more appropriate in general applications.)

7.5 Monte Carlo Results

A Monte Carlo analysis is conducted using the MMAE and Schruben's truncation-point selection algorithms. Four simulation models are used to generate the output sequences. For each of 1,000 sets, 10 replications of 1,500 observations each are generated.

The simulation models used to test this algorithm are simulations of open queuing systems and closed queuing systems. An open queuing system has new entities arriving, being processed, and departing. The multivariate output responses are a periodic sampling of the queue lengths at each server. If the queues are $M/M/1$ queues in tandem, Ross [79:326-330] shows that the queue lengths are independent and that the expected queue lengths for each server are the same as for stand-alone $M/M/1$ queues.

In contrast to the open queuing system, a closed queuing system has a fixed number of entities cycling through a circle of servers. The multivariate output responses are a periodic sampling of the number of customers at each server for all but one of the servers. If each

queue has exponentially distributed service times, Ross [79:330-335] provides the equations to calculate their analytic steady-state means.

The third type of multivariate output is from the open queuing model. Using two queues, the multivariate output consists of the the waiting time plus service time at the first server and the total system time. In contrast to the two previous cases which are "statistics based on time-persistent variables", these output sequences are "statistics based on observations".

The evaluation measures are based on the multivariate method of replications. On the truncated sequences, the confidence region is calculated with Equation (4) and associated volume with Equation (5). Over 1,000 sets of runs, the coverage rate for the analytic mean and average confidence region volumes are reported.

For comparison, Schruben's multivariate truncation algorithm, shown in Equations (14), (15), and (16), also is applied with $\alpha = 0.1$. For the sequences truncated with Schruben's algorithm, the confidence regions are calculated. The associated coverage rates and confidence region volumes are summarized.

The first simulation model is an open model of three $M/M/1$ queues in tandem. The interarrival time to the first queue in the series is exponentially distributed with a mean of 1.0. The queues have exponentially distributed service times of 0.5, 0.7, and 0.9 time units, respectively. Ross [79:326-330] shows that the length of queue for each server is the same as for an $M/M/1$ queue and that the queue lengths are independent of each other. The responses are the queue lengths with analytic steady-state solutions of $(0.5, 1.63, 8.1)^T$. Two sets of initial conditions are tested. The first is all empty queues and idle servers. The second set of initial conditions for this model is that of idle servers but with 26 customers in the first queue. In the first test, the system builds to steady state. In contrast, in the second test, the system starts overcongested, and the output sequences decays to their steady-state levels. The output responses are the queue lengths measured at every time interval for 1,500 observations.

Table 109. Open Model Queue Lengths Parameter Estimation Summary

Model	μ_{y_1} \bar{y}_1			μ_{y_2} \bar{y}_2			μ_{y_3} \bar{y}_3		
	$\hat{\mu}_{\bar{y}_1}$	$\hat{\sigma}_{\bar{y}_1}$		$\hat{\mu}_{\bar{y}_2}$	$\hat{\sigma}_{\bar{y}_2}$		$\hat{\mu}_{\bar{y}_3}$	$\hat{\sigma}_{\bar{y}_3}$	
Open Model (Empty and Idle)	0.50	0.50	0.03	1.63	1.64	0.15	8.10	8.09	1.71
Open Model ($L_{q_1}(t_0) = 26$)	0.50	0.50	0.03	1.63	1.64	0.15	8.10	8.12	1.74

Figure 21. Average Estimated Covariance Matrix $\hat{\Sigma}_{yy}$ ($\hat{\sigma}_{\hat{\Sigma}_{y_r, y_c}}$)

$$\begin{bmatrix} 0.124 (0.021) & -0.003 (0.023) & -0.018 (0.082) \\ -0.003 (0.023) & 0.638 (0.170) & -0.088 (0.334) \\ -0.018 (0.082) & -0.088 (0.334) & 5.874 (3.333) \end{bmatrix}$$

The steady-state model is estimated with the last half of the sequence. Over the 1,000 Monte Carlo sets of runs, Table 109 shows that the estimated steady-state means are very near the analytic solutions. Figure 21 shows the average covariance matrix and standard deviation (not the standard error) for the 1,000 sets with empty and idle initial conditions. The diagonal elements indicate that the queue length variance increases as traffic intensity increases. The off-diagonal elements are near zero because of the independence between queue lengths. Figures 22 and 23 show the average autocovariance matrices at one and two lags, respectively. The queue lengths exhibit positive autocovariances and insignificant cross-covariances.

Figures 24 and 25 show the average state transition matrices Φ and the average Kalman

Figure 22. Average Estimated Autocovariance Matrix at One Lag $\hat{\Gamma}_{yy}(1)$ ($\hat{\sigma}_{\hat{\Gamma}_{y_r, y_c}(1)}$)

$$\begin{bmatrix} 0.087 (0.020) & 0.013 (0.023) & -0.015 (0.082) \\ 0.013 (0.023) & 0.577 (0.169) & -0.058 (0.333) \\ -0.015 (0.082) & -0.058 (0.333) & 5.788 (3.334) \end{bmatrix}$$

Figure 23. Average Estimated Autocovariance Matrix at Two Lags $\hat{\Gamma}_{yy}(2) (\hat{\sigma}_{\hat{\Gamma}_{yy}(2)})$

$$\begin{bmatrix} 0.064 (0.019) & 0.021 (0.022) & -0.011 (0.081) \\ 0.021 (0.022) & 0.525 (0.167) & -0.033 (0.332) \\ -0.011 (0.081) & -0.033 (0.332) & 5.705 (3.334) \end{bmatrix}$$

Figure 24. Average State Transition Matrix $\Phi (\hat{\sigma}_{\Phi_{r,c}})$

$$\begin{bmatrix} 0.713 (0.052) & 0.023 (0.012) & 0.000 (0.004) \\ 0.023 (0.012) & 0.899 (0.025) & 0.005 (0.006) \\ 0.000 (0.004) & 0.005 (0.006) & 0.982 (0.010) \end{bmatrix}$$

filter gains K for the 1,000 Monte Carlo sets of runs. Because of the small cross-correlations, the off-diagonal terms are near zero. The diagonal elements of the state transition matrix show that the weights for propagating the estimates increase with the traffic intensity of the queues. Similarly, the Kalman filter gains increase with traffic intensity. The diagonal elements of the Kalman filter are relatively near their upper bound of 1.0. Since there are only insignificant differences between the average autocovariance matrices for the two initial conditions, only the average matrices are shown for the empty and idle initial conditions.

Table 110 shows statistics on the truncation points selected by the MMAE approach and by Schruben's approach. For these tests, MMAE selected a point in 1,999 out of 2,000 sets, whereas Schruben's algorithm failed to select a truncation point for 6 percent of the sets. The MMAE selected truncation points are considerably earlier in the sequences and

Figure 25. Average Kalman Filter Gain $K (\hat{\sigma}_{K_{r,c}})$

$$\begin{bmatrix} 0.694 (0.042) & 0.024 (0.011) & -0.001 (0.004) \\ 0.024 (0.011) & 0.897 (0.023) & 0.005 (0.006) \\ -0.001 (0.004) & 0.005 (0.006) & 0.981 (0.010) \end{bmatrix}$$

Table 110. Truncation Points \hat{n}_o

	Fail	Min	Max	Avg	St Dev
Open Model (Empty and Idle)					
MMAE Algorithm	0.001	24	1064	173.5	147.6
Schruben Algorithm	0.060	20	1480	353.5	355.1
Open Model ($L_{q_1}(t_0) = 26$)					
MMAE Algorithm	0.000	59	928	276.9	158.7
Schruben Algorithm	0.064	51	1481	447.2	351.0

Table 111. Truncation Point \hat{n}_o Evaluations

	\hat{n}_o	Cov	Vol
Open Model (Empty and Idle)			
MMAE Algorithm	173.5 ± 7.68	0.83 ± 0.020	0.44 ± 0.026
Schruben Algorithm	353.5 ± 19.05	0.82 ± 0.021	1.76 ± 0.614
Open Model ($L_{q_1}(t_0) = 26$)			
MMAE Algorithm	276.9 ± 8.25	0.85 ± 0.019	0.47 ± 0.019
Schruben Algorithm	447.2 ± 18.87	0.82 ± 0.020	1.58 ± 0.485

have less variance than Schruben's technique. Table 111 shows that the MMAE-selected truncation points result in sequences that achieve a higher coverage rate at a smaller average confidence region volume.

Table 112 depicts the average periodic output values when the initial conditions are empty and idle. The output increases to the steady-state values. The double horizontal line indicates where the average MMAE selected truncation point occurs, and the single horizontal line is where the average truncation point selected by Schruben's algorithm occurs. Table 113 shows periodic output values for the multivariate sequence when the initial queue length at the first server is 26. Since the first queue is initially overcongested, each of the three queues becomes overcongested during the transient. The truncated sequences resulting from both algorithms retain some transient data, particularly with the initial queue of 26 entities at the first server.

Table 112. Periodic Output Values Points for Open Model (Empty and Idle)

n	$\hat{\mu}_{y_{n,1}}$	$\hat{\sigma}_{y_{n,1}}$	$\hat{\mu}_{y_{n,2}}$	$\hat{\sigma}_{y_{n,2}}$	$\hat{\mu}_{y_{n,3}}$	$\hat{\sigma}_{y_{n,3}}$
μ_y	0.50		1.63		8.10	
1	0.29	0.68	0.16	0.43	0.03	0.18
100	0.48	1.09	1.60	2.54	5.53	5.93
200	0.50	1.13	1.65	2.55	6.77	7.38
300	0.50	1.12	1.60	2.57	7.37	8.32
400	0.50	1.12	1.60	2.50	7.57	8.71
500	0.49	1.08	1.63	2.55	7.77	8.96
600	0.51	1.12	1.63	2.59	7.99	9.18
700	0.50	1.12	1.60	2.54	8.12	9.34
800	0.48	1.09	1.61	2.55	8.06	9.33
900	0.48	1.10	1.64	2.55	7.98	9.29
1000	0.47	1.08	1.65	2.55	7.95	9.24
1100	0.52	1.15	1.63	2.59	7.95	9.24
1200	0.51	1.14	1.63	2.59	8.10	9.35
1300	0.48	1.07	1.62	2.57	8.26	9.49
1400	0.50	1.12	1.63	2.52	8.13	9.53
1500	0.50	1.13	1.65	2.60	8.26	9.62

Table 113. Periodic Output Values Points for Open Model ($L_{q_1}(t_0) = 26$)

n	$\hat{\mu}_{y_{n,1}}$	$\hat{\sigma}_{y_{n,1}}$	$\hat{\mu}_{y_{n,2}}$	$\hat{\sigma}_{y_{n,2}}$	$\hat{\mu}_{y_{n,3}}$	$\hat{\sigma}_{y_{n,3}}$
μ_y	0.50		1.63		8.10	
1	24.02	1.72	0.64	1.03	0.11	0.36
100	0.49	1.08	2.29	3.89	15.65	11.22
200	0.51	1.11	1.69	2.60	11.79	12.29
300	0.48	1.09	1.63	2.58	9.91	11.52
400	0.48	1.06	1.62	2.52	9.32	10.94
500	0.49	1.07	1.59	2.51	8.76	10.43
600	0.51	1.14	1.63	2.61	8.61	10.22
700	0.49	1.12	1.60	2.53	8.53	9.94
800	0.49	1.10	1.65	2.61	8.31	9.76
900	0.51	1.12	1.60	2.56	8.17	9.47
1000	0.50	1.11	1.66	2.57	7.97	9.33
1100	0.51	1.11	1.61	2.55	7.99	9.29
1200	0.50	1.13	1.62	2.53	8.13	9.35
1300	0.51	1.13	1.58	2.49	8.12	9.41
1400	0.48	1.10	1.63	2.53	8.14	9.30
1500	0.51	1.13	1.63	2.56	8.14	9.36

Table 114. Periodic Truncation Points for Open Model (Empty and Idle)

\hat{n}_0	$\hat{\mu}_y$			Cov	Vol
μ_y	0.50	1.63	8.10		
0	0.50	1.62	7.52	0.79 ± 0.02	0.30 ± 0.01
100	0.50	1.63	7.80	0.82 ± 0.02	0.35 ± 0.01
200	0.50	1.63	7.92	0.83 ± 0.02	0.40 ± 0.02
300	0.50	1.63	7.99	0.84 ± 0.02	0.45 ± 0.02
400	0.50	1.63	8.03	0.83 ± 0.02	0.51 ± 0.02
500	0.50	1.64	8.07	0.83 ± 0.02	0.58 ± 0.02
600	0.50	1.64	8.09	0.85 ± 0.02	0.67 ± 0.03
700	0.50	1.64	8.09	0.85 ± 0.02	0.79 ± 0.03
800	0.50	1.64	8.09	0.83 ± 0.02	0.94 ± 0.04
900	0.50	1.64	8.10	0.83 ± 0.02	1.15 ± 0.05
1000	0.50	1.64	8.14	0.82 ± 0.02	1.43 ± 0.06
1100	0.50	1.64	8.19	0.81 ± 0.02	1.88 ± 0.08
1200	0.50	1.64	8.22	0.80 ± 0.02	2.65 ± 0.11
1300	0.50	1.64	8.24	0.77 ± 0.02	4.27 ± 0.19
1400	0.50	1.64	8.23	0.77 ± 0.02	8.79 ± 0.45

Table 114 shows for the periodic truncated sequences, the average estimated responses along with the corresponding coverage rate and confidence region volume for the open model with empty and idle initial conditions. Table 115 shows the same statistics when the initial conditions are 26 customers at the first server. The estimated responses converge to the analytic values by the 800th observation. However, the mean estimator error is very small considerably earlier in the sequences. The coverage rate begins at zero with no truncation, but is very near the nominal rate of 0.9 after truncating only 100 observations. The coverage rate continues to decrease with additional truncation. The confidence region volume initially decreases, probably because of the high variability in the first 100 observations. Additional deletions after the first 100 data points result in increasing the confidence region volume. The average MMAE truncation point is indicated with double horizontal bars, and the average truncation point selected by Schruben's algorithm is shown with a single bar.

Table 115. Periodic Truncation Points for Open Model ($L_{q1}(t_0) = 26$)

\hat{n}_0	$\hat{\mu}_y$			Cov	Vol
μ_y	0.50	1.63	8.10		
0	0.71	1.97	9.15	0.00 ± 0.00	0.85 ± 0.03
100	0.50	1.64	8.95	0.89 ± 0.02	0.45 ± 0.02
200	0.50	1.63	8.60	0.88 ± 0.02	0.47 ± 0.02
300	0.50	1.63	8.41	0.87 ± 0.02	0.51 ± 0.02
400	0.50	1.63	8.30	0.86 ± 0.02	0.56 ± 0.02
500	0.50	1.64	8.23	0.85 ± 0.02	0.62 ± 0.03
600	0.50	1.64	8.18	0.84 ± 0.02	0.70 ± 0.03
700	0.50	1.64	8.14	0.83 ± 0.02	0.80 ± 0.04
800	0.50	1.64	8.10	0.82 ± 0.02	0.94 ± 0.05
900	0.50	1.64	8.07	0.81 ± 0.02	1.13 ± 0.05
1000	0.50	1.64	8.07	0.81 ± 0.02	1.40 ± 0.06
1100	0.50	1.64	8.09	0.81 ± 0.02	1.85 ± 0.08
1200	0.50	1.64	8.10	0.79 ± 0.02	2.57 ± 0.11
1300	0.50	1.64	8.09	0.77 ± 0.02	4.08 ± 0.18
1400	0.50	1.65	8.08	0.76 ± 0.02	8.24 ± 0.41

The next model is a closed network of servers. Four servers, each with exponentially-distributed service times with means of 0.8, 0.7, 0.6, and 0.5, are arranged in a ring. Thirty customers, initially at the second queue, are processed around the ring. The system is closed, so by definition no additional customers arrive and no current customers leave. The responses are the number of customers in line or being served for the first three servers. (If all four queues are used, the multivariate output is linearly dependent.) Ross [79:330-335] provides an algorithm to calculate the analytic steady-state value of the response. For this system, the vector of steady-state responses is $(19.33, 6.14, 2.89)^T$. Beginning with all the customers queued at the second server induces a significant transient. The responses are sampled every time unit until 1,500 observations are recorded.

Table 116 shows that, for the 1,000 Monte Carlo sets of runs, the mean responses based on the last half of the output sequences are very near the analytic values. Figures 26 through 28 shows the average estimated covariance matrix and autocovariance matrices.

Table 116. Open Model (4 Servers) Queue Lengths Parameter Estimation Summary

Model	μ_{y_1}	\bar{y}_1		μ_{y_2}	\bar{y}_2		μ_{y_3}	\bar{y}_3	
		$\hat{\mu}_{\bar{y}_1}$	$\hat{\sigma}_{\bar{y}_1}$		$\hat{\mu}_{\bar{y}_2}$	$\hat{\sigma}_{\bar{y}_2}$		$\hat{\mu}_{\bar{y}_3}$	$\hat{\sigma}_{\bar{y}_3}$
Closed Model (4 Servers)	19.33	19.31	0.65	6.14	6.16	0.54	2.89	2.89	0.19

Figure 26. Average Estimated Covariance Matrix $\hat{\Sigma}_{yy} (\hat{\sigma}_{\Sigma_{y_r, y_c}})$

$$\begin{bmatrix} 4.114 (1.277) & -2.892 (1.055) & -0.868 (0.371) \\ -2.892 (1.055) & 3.087 (0.971) & -0.142 (0.264) \\ -0.868 (0.371) & -0.142 (0.264) & 1.025 (0.228) \end{bmatrix}$$

Each of these matrices shows a negative covariance with the number of customers at other queues. Since the number of customers is fixed, this negative cross-correlation is logical.

Figure 29 shows the average state transition matrix and standard deviations over the 1,000 sets. In propagating the estimates of the number at each server, the previous number at that server has the largest weight, with small negative weights assigned to the other server queue lengths. The average Kalman filter gains are shown in Figure 30. The diagonal elements are relatively close to 1.0.

Table 117 shows the results of applying both the MMAE and Schruben's truncation point selection algorithms. With highly interdependent responses, Schruben's algorithm selected a truncation point only 32 times out of the 1,000 sets of runs. In contrast, the MMAE approach selected a truncation point every run. Table 118 shows that the coverage

Figure 27. Average Estimated Autocovariance Matrix at One Lag $\hat{\Gamma}_{yy}(1) (\hat{\sigma}_{\Gamma_{y_r, y_c}(1)})$

$$\begin{bmatrix} 3.990 (1.277) & -2.833 (1.054) & -0.851 (0.371) \\ -2.833 (1.054) & 2.970 (0.970) & -0.092 (0.264) \\ -0.851 (0.371) & -0.092 (0.264) & 0.918 (0.227) \end{bmatrix}$$

Figure 28. Average Estimated Autocovariance Matrix at Two Lags $\hat{\Gamma}_{yy}(2)$ ($\hat{\sigma}_{\hat{\Gamma}_{yy}(2)}$)

$$\begin{bmatrix} 3.868 (1.275) & -2.773 (1.053) & -0.821 (0.370) \\ -2.773 (1.053) & 2.865 (0.969) & -0.056 (0.263) \\ -0.821 (0.370) & -0.056 (0.263) & 0.832 (0.225) \end{bmatrix}$$

Figure 29. Average State Transition Matrix Φ ($\hat{\sigma}_{\Phi}$)

$$\begin{bmatrix} 0.892 (0.028) & -0.082 (0.027) & -0.072 (0.031) \\ -0.026 (0.023) & 0.936 (0.025) & 0.012 (0.027) \\ -0.073 (0.025) & -0.062 (0.026) & 0.825 (0.037) \end{bmatrix}$$

Figure 30. Average Kalman Filter Gain K ($\hat{\sigma}_{K}$)

$$\begin{bmatrix} 0.877 (0.022) & -0.097 (0.021) & -0.098 (0.022) \\ 0.000 (0.000) & 0.946 (0.019) & 0.032 (0.019) \\ 0.000 (0.000) & 0.000 (0.000) & 0.796 (0.031) \end{bmatrix}$$

Table 117. Truncation Points \hat{n}_o

	Fail	Min	Max	Avg	St Dev
Closed Model (4 Servers)					
MMAE Algorithm	0.000	55	344	113.9	36.5
Schruben Algorithm	0.968	70	1460	619.1	430.9

Table 118. Truncation Point \hat{n}_0 Evaluations

	\hat{n}_0	Cov	Vol
Closed Model (4 Servers)			
MMAE Algorithm	113.9 ± 1.90	0.88 ± 0.017	0.38 ± 0.010
Schruben Algorithm	619.1 ± 125.31	0.81 ± 0.114	1.64 ± 0.858

Table 119. Periodic Output Values Points for the Closed Model (4 Servers)

n	$\hat{\mu}_{y_{n,1}}$	$\hat{\sigma}_{y_{n,1}}$	$\hat{\mu}_{y_{n,2}}$	$\hat{\sigma}_{y_{n,2}}$	$\hat{\mu}_{y_{n,3}}$	$\hat{\sigma}_{y_{n,3}}$
μ_y	19.33		6.14		2.89	
1	0.18	0.43	28.66	1.19	0.84	1.04
100	16.73	7.66	7.89	6.85	3.50	3.84
200	19.14	6.90	6.29	6.01	2.93	3.29
300	19.27	6.80	6.22	5.93	2.88	3.29
400	19.33	6.74	6.10	5.80	2.94	3.30
500	19.33	6.85	6.14	5.91	2.89	3.25
600	19.38	6.74	6.08	5.79	2.93	3.36
700	19.27	6.83	6.16	5.88	2.89	3.23
800	19.25	6.79	6.24	5.83	2.87	3.24
900	19.25	6.84	6.16	5.88	2.92	3.28
1000	19.29	6.70	6.15	5.88	2.94	3.23
1100	19.45	6.71	6.01	5.78	2.88	3.23
1200	19.19	6.81	6.26	5.90	2.89	3.27
1300	19.36	6.72	6.16	5.86	2.83	3.23
1400	19.29	6.78	6.18	5.88	2.88	3.24
1500	19.25	6.85	6.16	5.89	2.93	3.33

rate for the sequences truncated at the MMAE selected point is very near nominal. The coverage rate and volume for Schruben's algorithm are based only on the sequences resulting from the 32 selected truncation points.

Table 119 shows periodic output responses, and Table 120 shows the estimated responses, coverage rates, and confidence region volume for periodic truncation points. These two tables indicate that the MMAE average selected truncation point of 113.9 is both effective in terms of removing the transient and efficient by not excessively truncating the

Table 120. Periodic Truncation Points for the Closed Model (4 Servers)

\hat{n}_0	$\hat{\mu}_y$			Cov	Vol
μ_y	19.33	6.14	2.89		
0	18.62	6.73	2.97	0.78 ± 0.02	0.36 ± 0.01
100	19.24	6.21	2.91	0.90 ± 0.02	0.38 ± 0.01
200	19.31	6.16	2.89	0.88 ± 0.02	0.42 ± 0.01
300	19.31	6.16	2.89	0.88 ± 0.02	0.46 ± 0.01
400	19.31	6.16	2.89	0.87 ± 0.02	0.52 ± 0.01
500	19.32	6.16	2.89	0.88 ± 0.02	0.60 ± 0.02
600	19.32	6.16	2.89	0.88 ± 0.02	0.70 ± 0.02
700	19.31	6.16	2.89	0.88 ± 0.02	0.83 ± 0.02
800	19.31	6.16	2.89	0.89 ± 0.02	0.99 ± 0.03
900	19.32	6.16	2.89	0.87 ± 0.02	1.24 ± 0.04
1000	19.32	6.16	2.88	0.89 ± 0.02	1.62 ± 0.05
1100	19.31	6.17	2.88	0.86 ± 0.02	2.21 ± 0.07
1200	19.31	6.17	2.88	0.87 ± 0.02	3.26 ± 0.11
1300	19.30	6.18	2.89	0.83 ± 0.02	5.69 ± 0.21
1400	19.26	6.21	2.90	0.81 ± 0.02	13.11 ± 0.57

sequences.

The closed model is also run with five servers. The mean service times are 0.8, 0.7, 0.6, 0.5 and 0.4 time units. The analytic numbers of entities at each of the first four servers are $(18.46, 6.05, 2.88, 1.63)^T$. Table 121 demonstrates that these values are estimated fairly closely with the last half of the output sequences. Figures 31 through 33 show the positive autocovariances and the negative covariances among the responses. Figures 34 and 35 show the average state transition matrix and average Kalman filter gain. Since the off-diagonal elements in the state transition matrix and Kalman filter gains are small on the average, the Kalman filter relies mostly on measurements of each response for estimating the corresponding state.

Statistics on the truncation points are shown in Table 122 for both the MMAE and Schruber's algorithms. Schruber's algorithm selected a truncation point in only 3 of the 1,000 sets of runs. Schruber's approach appears not to work well with negatively correlated

Table 121. Closed Model (5 Servers) Parameter Estimation Summary

Model	μ_{y_1} \bar{y}_1			μ_{y_2} \bar{y}_2			μ_{y_3} \bar{y}_3			μ_{y_4} \bar{y}_4		
		$\hat{\mu}_{\bar{y}_1}$	$\hat{\sigma}_{\bar{y}_1}$		$\hat{\mu}_{\bar{y}_2}$	$\hat{\sigma}_{\bar{y}_2}$		$\hat{\mu}_{\bar{y}_3}$	$\hat{\sigma}_{\bar{y}_3}$		$\hat{\mu}_{\bar{y}_4}$	$\hat{\sigma}_{\bar{y}_4}$
Closed Model	18.46	18.48	0.63	6.05	6.03	0.52	2.88	2.88	0.19	1.63	1.63	0.08

Figure 31. Average Estimated Covariance Matrix $\hat{\Sigma}_{yy} (\hat{\sigma}_{\hat{\Sigma}_{y_r, y_c}})$

$$\begin{bmatrix} 4.093 (1.262) & -2.732 (1.000) & -0.851 (0.375) & -0.350 (0.147) \\ -2.732 (1.000) & 2.966 (0.899) & -0.157 (0.258) & -0.055 (0.115) \\ -0.851 (0.375) & -0.157 (0.258) & 1.026 (0.235) & -0.012 (0.063) \\ -0.350 (0.147) & -0.055 (0.115) & -0.012 (0.063) & 0.419 (0.068) \end{bmatrix}$$

Figure 32. Average Estimated Autocovariance Matrix at One Lag $\hat{\Gamma}_{yy}(1) (\hat{\sigma}_{\hat{\Gamma}_{y_r, y_c}(1)})$

$$\begin{bmatrix} 3.969 (1.262) & -2.673 (0.999) & -0.843 (0.374) & -0.332 (0.147) \\ -2.673 (0.999) & 2.850 (0.898) & -0.107 (0.257) & -0.048 (0.115) \\ -0.843 (0.374) & -0.107 (0.257) & 0.919 (0.234) & 0.028 (0.062) \\ -0.332 (0.147) & -0.048 (0.115) & 0.028 (0.062) & 0.324 (0.066) \end{bmatrix}$$

Figure 33. Average Estimated Autocovariance Matrix at Two Lags $\hat{\Gamma}_{yy}(2) (\hat{\sigma}_{\hat{\Gamma}_{y_r, y_c}(2)})$

$$\begin{bmatrix} 3.846 (1.260) & -2.618 (0.999) & -0.822 (0.373) & -0.305 (0.146) \\ -2.618 (0.999) & 2.745 (0.897) & -0.072 (0.256) & -0.038 (0.114) \\ -0.822 (0.373) & -0.072 (0.256) & 0.834 (0.231) & 0.048 (0.062) \\ -0.305 (0.146) & -0.038 (0.114) & 0.048 (0.062) & 0.261 (0.064) \end{bmatrix}$$

Figure 34. Average State Transition Matrix Φ ($\hat{\sigma}_{\Phi_{r,c}}$)

$$\begin{bmatrix} 0.823 (0.056) & -0.151 (0.055) & -0.151 (0.056) & -0.101 (0.068) \\ -0.020 (0.045) & 0.941 (0.046) & 0.016 (0.047) & 0.003 (0.056) \\ -0.075 (0.046) & -0.063 (0.046) & 0.823 (0.053) & -0.007 (0.058) \\ -0.112 (0.045) & -0.110 (0.046) & -0.082 (0.046) & 0.670 (0.070) \end{bmatrix}$$

Figure 35. Average Kalman Filter Gain \mathbf{K} ($\hat{\sigma}_{\mathbf{K}_{r,c}}$)

$$\begin{bmatrix} 0.784 (0.029) & -0.190 (0.029) & -0.200 (0.029) & -0.165 (0.031) \\ 0.000 (0.000) & 0.955 (0.024) & 0.041 (0.024) & 0.008 (0.026) \\ 0.000 (0.000) & 0.000 (0.000) & 0.846 (0.033) & 0.052 (0.030) \\ 0.000 (0.000) & 0.000 (0.000) & 0.000 (0.000) & 0.614 (0.041) \end{bmatrix}$$

Table 122. Truncation Points \hat{n}_o

	Fail	Min	Max	Avg	St Dev
Closed Model (5 Servers)					
MMAE Algorithm	0.000	41	301	105.5	32.7
Schruben Algorithm	0.997	500	1290	893.3	322.5

Table 123. Truncation Point \hat{n}_0 Evaluations

	\hat{n}_0	Cov	Vol
Closed Model (5 Servers)			
MMAE Algorithm	105.5 ± 1.70	0.85 ± 0.018	0.07 ± 0.002
Schruben Algorithm	893.3 ± 543.73	0.33 ± 0.795	1.25 ± 2.581

Table 124. Periodic Output Values Points for Closed Model (5 Servers)

n	$\hat{\mu}_{y_{n,1}}$	$\hat{\sigma}_{y_{n,1}}$	$\hat{\mu}_{y_{n,2}}$	$\hat{\sigma}_{y_{n,2}}$	$\hat{\mu}_{y_{n,3}}$	$\hat{\sigma}_{y_{n,3}}$	$\hat{\mu}_{y_{n,4}}$	$\hat{\sigma}_{y_{n,4}}$
μ_y	18.46		6.05		2.88		1.63	
1	0.08	0.29	28.59	1.19	0.85	1.03	0.33	0.59
100	16.40	7.48	7.43	6.54	3.29	3.68	1.79	2.25
200	18.35	6.84	6.15	5.87	2.87	3.21	1.64	2.04
300	18.28	6.82	6.14	5.80	2.94	3.37	1.67	2.16
400	18.36	6.80	6.14	5.77	2.89	3.25	1.63	2.00
500	18.30	6.81	6.11	5.78	2.96	3.38	1.66	2.10
600	18.45	6.79	6.15	5.76	2.79	3.23	1.59	2.04
700	18.47	6.74	5.99	5.68	2.93	3.27	1.63	2.06
800	18.41	6.78	6.10	5.77	2.86	3.30	1.64	2.07
900	18.47	6.64	6.04	5.60	2.86	3.25	1.64	2.05
1000	18.42	6.74	6.14	5.76	2.88	3.23	1.59	2.03
1100	18.53	6.70	5.98	5.64	2.87	3.28	1.63	2.09
1200	18.44	6.75	6.01	5.69	2.89	3.22	1.68	2.08
1300	18.45	6.69	6.04	5.68	2.88	3.23	1.65	2.07
1400	18.47	6.74	6.00	5.70	2.92	3.32	1.63	2.06
1500	18.44	6.70	5.97	5.66	2.93	3.31	1.65	2.07

output responses. The MMAE algorithm truncated 105.5 observations on the average. The MMAE truncated sequences result in a coverage rate of 85 percent and an average confidence region volume of 0.07. Only one of three truncated points selected by Schruben's routine result in a confidence region which covered the analytic mean.

The table of periodic output values, Table 124, shows that large transient effects last less than 200 observations, but small transient effects are seen until about the 600th observation. From Table 125, the transient is insignificant on the confidence region coverage rate

Table 125. Periodic Truncation Points for Closed Model (5 Servers)

\hat{n}_0	$\hat{\mu}_y$				Cov	Vol
μ_y	18.46	6.05	2.88	1.63		
0	17.80	6.58	2.95	1.65	0.78 ± 0.02	0.06 ± 0.00
100	18.38	6.10	2.89	1.64	0.88 ± 0.02	0.07 ± 0.00
200	18.43	6.06	2.88	1.63	0.89 ± 0.02	0.08 ± 0.00
300	18.44	6.06	2.88	1.63	0.87 ± 0.02	0.09 ± 0.00
400	18.45	6.05	2.88	1.63	0.87 ± 0.02	0.10 ± 0.00
500	18.46	6.04	2.88	1.63	0.88 ± 0.02	0.12 ± 0.00
600	18.47	6.03	2.88	1.63	0.89 ± 0.02	0.15 ± 0.01
700	18.48	6.03	2.88	1.63	0.89 ± 0.02	0.19 ± 0.01
800	18.48	6.03	2.87	1.63	0.90 ± 0.02	0.25 ± 0.01
900	18.48	6.03	2.88	1.63	0.88 ± 0.02	0.33 ± 0.01
1000	18.48	6.02	2.88	1.63	0.86 ± 0.02	0.47 ± 0.02
1100	18.49	6.02	2.87	1.63	0.87 ± 0.02	0.72 ± 0.03
1200	18.48	6.03	2.87	1.63	0.85 ± 0.02	1.23 ± 0.05
1300	18.49	6.02	2.88	1.63	0.85 ± 0.02	2.60 ± 0.11
1400	18.48	6.01	2.88	1.64	0.81 ± 0.02	8.31 ± 0.41

after truncating 100 observations.

The final output considered is the waiting and service times for entities in a simulation of two tandem $M/M/1$ queues. Both the interarrival and service times are exponentially distributed. The mean interarrival time is 1.0 time unit, and the means of the service times are 0.6 and 0.8, respectively. Ross [79] shows that the queues are independent with the analytic solutions the same as individual $M/M/1$ queues. The first response is the waiting time plus service time at the first server, and the second response is total time in the system. Since the second response is the sum the first response plus an independent random variable, the responses are moderately correlated.

This output sequence is an example of "statistics based on observations", whereas the previous sequences are "statistics based on time-persistent variables". In this case, the state transition matrix represents the relationship between subsequent entities rather than the relationship over a fixed time interval. The state transition is estimated in the same way

Table 26. Open Model Waiting Times Parameter Estimation Summary

Model	μ_{y_1} \bar{y}_1			μ_{y_2} \bar{y}_2		
		$\hat{\mu}_{\bar{y}_1}$	$\hat{\sigma}_{\bar{y}_1}$		$\hat{\mu}_{\bar{y}_2}$	$\hat{\sigma}_{\bar{y}_2}$
Open Model Waiting Times	1.50	1.50	0.07	5.50	5.49	0.44

Figure 36. Average Estimated Covariance Matrix $\hat{\Sigma}_{yy}$ ($\hat{\sigma}_{\hat{\Sigma}_{y_r, y_c}}$)

$$\begin{bmatrix} 0.223 (0.045) & 0.216 (0.090) \\ 0.216 (0.090) & 1.659 (0.654) \end{bmatrix}$$

and is still held constant in the Kalman filter.

The steady-state models are estimated using the last half of the output sequences. Table 126 shows that the average steady-state mean estimates are centered on the analytic means. Figures 36 through 38 indicate that the output sequences have positive covariances and positive autocovariances.

Figure 39 shows the average state transition matrix, and Figure 40 shows the average Kalman filter gains. Since the off-diagonal elements are near zero, the MMAE approach does not rely on the covariance between the responses.

The MMAE algorithm never failed to select a truncation point, while Schruben's approach failed in 4.3 percent of the tests, as shown in Table 127. The MMAE truncation points are considerably earlier in the sequence, with much less variance than the truncation points selected by Schruben's algorithm. Table 128 shows that the resulting coverage

Figure 37. Average Estimated Autocovariance Matrix at One Lag $\hat{\Gamma}_{yy}(1)$ ($\hat{\sigma}_{\hat{\Gamma}_{y_r, y_c}(1)}$)

$$\begin{bmatrix} 0.187 (0.045) & 0.192 (0.090) \\ 0.192 (0.090) & 1.584 (0.654) \end{bmatrix}$$

Figure 38. Average Estimated Autocovariance Matrix at Two Lags $\hat{\Gamma}_{yy}(2)$ ($\hat{\sigma}_{\hat{\Gamma}_{yy}(2)}$)

$$\begin{bmatrix} 0.160 (0.044) & 0.174 (0.090) \\ 0.174 (0.090) & 1.517 (0.653) \end{bmatrix}$$

Figure 39. Average State Transition Matrix Φ ($\hat{\sigma}_{\Phi_{r,c}}$)

$$\begin{bmatrix} 0.838 (0.034) & 0.007 (0.008) \\ -0.066 (0.031) & 0.959 (0.017) \end{bmatrix}$$

Figure 40. Average Kalman Filter Gain K ($\hat{\sigma}_{K_{r,c}}$)

$$\begin{bmatrix} 0.822 (0.032) & 0.008 (0.009) \\ 0.000 (0.000) & 0.958 (0.016) \end{bmatrix}$$

Table 127. Truncation Points \hat{n}_o

	Fail	Min	Max	Avg	St Dev
Open Model Waiting Times					
MMAE Algorithm	0.000	9	310	36.1	24.2
Schruben Algorithm	0.043	0	1480	236.3	283.3

Table 128. Truncation Point \hat{n}_0 Evaluations

	\hat{n}_0	Cov	Vol
Open Model Waiting Times			
MMAE Algorithm	36.1 ± 1.26	0.87 ± 0.017	0.26 ± 0.007
Schruben Algorithm	236.3 ± 15.06	0.88 ± 0.018	0.43 ± 0.057

Table 129. Periodic Output Values Points for Open Model Waiting Times

n	$\hat{\mu}_{y_{n,1}}$	$\hat{\sigma}_{y_{n,1}}$	$\hat{\mu}_{y_{n,2}}$	$\hat{\sigma}_{y_{n,2}}$
μ_y	1.50		5.50	
1	0.62	0.61	1.43	1.03
100	1.49	1.49	5.31	4.00
200	1.51	1.53	5.47	4.16
300	1.50	1.49	5.48	4.19
400	1.49	1.49	5.51	4.31
500	1.50	1.48	5.50	4.27
600	1.50	1.50	5.53	4.28
700	1.47	1.48	5.53	4.37
800	1.51	1.52	5.54	4.31
900	1.50	1.49	5.52	4.33
1000	1.49	1.49	5.43	4.22
1100	1.50	1.52	5.49	4.27
1200	1.50	1.51	5.54	4.31
1300	1.50	1.50	5.57	4.42
1400	1.49	1.50	5.48	4.25
1500	1.50	1.49	5.50	4.21

rates are nearly equal, but that the MMAE truncation point result in considerably smaller confidence region volumes.

Table 129 shows periodic responses. The analytic means are obtained at the 400th observation on average. The evaluation of periodic truncation points are given in Table 130. Very near nominal coverage rates are achieved with only small amounts of truncation. The MMAE approach which truncates early in the sequence appears preferable to Schruben's algorithm.

Table 130. Periodic Truncation Points for Open Model Waiting Times

\hat{n}_0	$\hat{\mu}_y$		Cov	Vol
μ_y	1.50	5.50		
0	1.50	5.43	0.86 ± 0.02	0.25 ± 0.01
100	1.50	5.49	0.88 ± 0.02	0.27 ± 0.01
200	1.50	5.50	0.87 ± 0.02	0.29 ± 0.01
300	1.50	5.50	0.87 ± 0.02	0.32 ± 0.01
400	1.50	5.50	0.89 ± 0.02	0.35 ± 0.01
500	1.50	5.50	0.87 ± 0.02	0.38 ± 0.01
600	1.50	5.50	0.87 ± 0.02	0.42 ± 0.01
700	1.50	5.50	0.86 ± 0.02	0.46 ± 0.01
800	1.50	5.49	0.86 ± 0.02	0.52 ± 0.02
900	1.50	5.49	0.85 ± 0.02	0.59 ± 0.02
1000	1.50	5.49	0.85 ± 0.02	0.69 ± 0.02
1100	1.50	5.50	0.84 ± 0.02	0.85 ± 0.03
1200	1.50	5.50	0.82 ± 0.02	1.09 ± 0.04
1300	1.50	5.49	0.82 ± 0.02	1.48 ± 0.06
1400	1.50	5.46	0.79 ± 0.02	2.45 ± 0.10

7.6 Summary

The multivariate MMAE truncation-point selection algorithm is very successful on the output of four simulation models. The truncated sequences have very near nominal coverage rates and relatively small confidence region volumes. The test cases include two cases of "statistics based on time-persistent variables": the open queuing system with uncorrelated responses and the closed queuing system with negatively correlated responses. The third test case is an example of "statistics based upon observations" from an open queuing system. In all of these cases, the sequences truncated with the MMAE points are more effective (in terms of coverage rates) and more efficient (in terms of confidence region volume) than sequences truncated with points selected by Schruben's [86] algorithm.

Schruben's method [86] rarely selected a truncation point for the closed queuing system. In comparison to the other cases, these responses have relatively stronger correlation and further have negative correlation. Schruben's method is based on the sample mean vector

and sample covariance matrix estimated with only the last batch from each observation. Since the recommended batches are small, only 5 to 10 observations, very little data is used for these estimates. The variance of these estimates may have contributed to the poor performance of his technique. In his article, Schruben [86] presents only one trial case for his algorithm. His example case is based on the time-sharing computer model studied by Adirir and Avi-Itzhak [3]. The responses are the computer response time, a "statistic based on observations" and a zero-one indicator variable of the cpu's busy/idle status, which is a "statistic based on time". Because the sequences are different types, how the data is collected in pairs is not obvious. Although not specified, it appears Schruben took these two separate univariate sequences and matched up corresponding terms to form his multivariate sequence. Since the pairs of responses may not have measured the system under the same conditions, the multivariate sequence may not have significant covariance. Schruben may have tested his multivariate algorithm under the conditions of little covariance. As seen in this chapter, his algorithm performs well with no covariance, but less satisfactorily with negative covariance.

VIII. Summary and Recommendations

8.1 Summary

The significance of this research is the introduction of new concepts to the field of discrete-event simulation output analysis. Specifically, modeling with measurement noise, applying Kalman filters, and using Multiple Model Adaptive Estimation (MMAE) are all new approaches to analyzing simulation output. As a result of applying these concepts, new algorithms for truncation-point selection for both univariate and multivariate output sequences are proposed and tested. These algorithms are applicable to both "statistics based on time-persistent variables" and "statistics based on observations". The truncation-point selection algorithms based on MMAE result in truncated sequences with better mean estimates and confidence intervals.

This research begins with a review of simulation output analysis. Particular focus is placed on the initial data bias or warm-up problem. An overview of stochastic processes is presented, leading to an introduction of the Kalman filter. The Kalman filter is a state-space stochastic estimation algorithm. Several approaches for system identification and estimation are reviewed.

Using the foundation of previous simulation research and stochastic estimation with the Kalman filter, two methods to select truncation points to eliminate the initial data bias are developed. The first truncation-point selection approach is based on residual monitoring using a single Kalman filter. The knowledge gained from this attempt laid the foundation for an improved approach based on Multiple Model Adaptive Estimation (MMAE). MMAE is an estimation technique that uses a bank of Kalman filters with different estimates. In the MMAE method, the output is estimated as an autoregressive model observed with measurement noise using the last half of the output sequence. The MMAE filters have different estimates for the output means. Based on the filters' residuals, a time-varying MMAE mean estimate is calculated. When the MMAE mean estimate is close to the mean estimator from

the last half of the output sequence for a predetermined number of observations, a truncation point is selected.

The MMAE truncation algorithm performs successfully on a single long run of simulation output. However, for single long runs, the effects of the initial transient on making confidence intervals may be negligible. The MMAE truncation algorithm is very successful with multiple replications for both univariate and multivariate output sequences. The selected truncation points effectively reduce the effects of the initial bias, without excessive deletion which increases the variance of the estimates.

8.2 Recommendations for Improvements to the MMAE Truncation-Point Selection Algorithms

Instead of the correlation technique, perhaps a different estimation approach for the steady-state parameters could be employed. Law and Kelton [53:286] show that autocorrelation estimates may be biased. If the sample size is fixed to the last half of the output sequence, MLE or least squares can be used instead of the correlation technique. More elaborate estimation schemes may be possible. For example, Minimum Distance Estimation has been found to outperform MLE for small static samples [25]. If the estimate of \hat{k} can be improved, the variance estimates may be more reliable and may prove useful.

A better scheme for determining the MMAE filter spacing or truncation point may be possible. The tested algorithm estimates the steady-state model and finds the average, minimum, and maximum for each response from the last half of the output sequences. The MMAE filters are positioned at each combination of the response averages, minimum, and maximum. The truncation point is selected when the MMAE time-varying mean estimate is within one of the Euclidean distance to the response averages for 5 consecutive observations. These results are based on an initial approach without attempting to optimize. Perhaps a different vector norm or consecutive number of observations may improve the results. At a minimum, a threshold for the difference in mean estimates in terms of the sample output

mean, rather than the absolute value of one, may make the algorithm applicable to a wider range of model outputs.

A transformation of the data, such as natural logarithms of each observation, may reduce the effect of the exponential output distribution from many queuing systems. If the output distribution is closer to the normal distribution, the Kalman filter assumptions of normally distributed noise terms may be more appropriate.

The underlying model could possibly be changed from an AR(2) with measurement noise model to regulated Brownian motion. Several researchers [1, 30, 102] have applied regulated Brownian motion models to simulation output with success. The resulting Kalman filter may have significant nonlinearities.

8.3 Potential Extensions to Other Discrete-Event Simulation Analysis Applications

Several other potential applications of Kalman filters for simulation output analysis, which not developed in this dissertation, are presented. These proposals are grouped into four categories: mean vector estimation, covariance estimation, variance reduction, and run length control. The univariate case is a special case of each multivariate proposal.

8.3.1 Multivariate Estimation of the Mean Vector. Develop an improved method of multivariate estimation of the mean vector with a joint confidence region for the estimated parameters. Three possibilities are apparent, but each of these concepts is extremely dependent upon the definition of the system states used in the Kalman filter.

The first idea for developing a confidence interval for the parameters is to relate the Kalman filter model estimates and covariance matrices to the estimated parameter's covariance matrix. In the process of running the Kalman filter, a covariance matrix of the Kalman filter states is developed. Along with this covariance matrix of states, the cross-covariance between the states and the parameters could be explored if nonlinear Kalman filter approaches (See Maybeck [58]) are applied. If this information could be related to the covariance matrix of the parameter estimates of the simulation output distributions, a new

technique for developing confidence intervals and regions can be tested. Another potential source of variance estimates are the probabilities from Multiple Model Adaptive Estimation [58:68-144]. These approaches are tested in the univariate case by Howard [37] and Howard, Gallagher, Bauer and Maybeck [39, 38].

The second concept addresses the problem of asynchronous output. Asynchronous output occurs when each output vector has only some of the responses [12, 90, 91]. If functions of the simulation output responses are included as Kalman filter states, the information from the asynchronous output could be incorporated using scalar measurement update techniques. (For scalar measurement updates, see Maybeck [57].) The Kalman filter algorithm may produce the estimated mean and covariance matrix of interest for the multivariate stochastic process. The difficulty is to define and estimate the state transition matrix Φ . Synder [95] may provide the necessary mathematical foundations.

Third, mean estimates are calculated currently by a weighted average of observation values, but the Kalman filter produces a sequence of state estimates. Perhaps, the Kalman filter state estimates may provide an improved estimate of the mean vector. Howard [37] finds there is very little difference between the average state estimate and the average of the data. This occurs for his applications because the Kalman filter gains are almost 1.0, which simply makes each state estimate, the propagation of the last two data values. With smaller Kalman filter gains, the state estimates may smooth the sequence by decreasing the magnitude of outliers.

8.3.2 Covariance Matrix Estimation. Besides estimating the variance of mean estimates, the covariance of the underlying multivariate stochastic process is also of interest. In fitting the Kalman filter to the simulation output, some estimate of the covariance of the underlying stochastic process is embedded in the analysis process. The MMAE technique [58:68-144] can also be used to estimate the covariance matrix. In the univariate case, this variance estimate can be compared with the Welch's [101:300-302] jackknife technique to estimate the system variance.

8.3.3 Improved Control Variates Applications. Variance reduction is a collection of techniques to reduce the variance of estimates. One specific class of variance reduction techniques is the method of control variates. Control variate methods use the covariance between the input pseudorandom control variates and the output sequence. Based on this covariance, conditional estimates of the parameters of interest are determined, given the actual realizations of the control variates. While the current techniques use the aggregate average of the control variates over an entire simulation run, the Kalman filter potentially can be used to develop the covariance based on the individual control variates and the affected entities. Since more information is used, better estimates with smaller variances may result.

8.3.4 Enhanced Simulation Run-Length Control. One difficulty with simulation output analysis is to determine the necessary run length to achieve a predetermined accuracy in the parameter estimates. Any information gained from the Kalman filter about the correlation among the output sequence or the stochastic process variance can be used to improve the current run length determination techniques. Perhaps a "rule of thumb" relating necessary simulation run length from the estimate of $\hat{\phi}_1$ can be developed.

Bibliography

1. Joseph Abate and Ward Whitt. Transient behavior of regulated brownian motion, I: Starting at the origin. *Advanced Applied Probability*, 19:560-598, 1987.
2. Bovas Abraham and Johannes Ledolter. *Statistical Methods for Forecasting*. John Wiley & Sons, Inc., New York, NY, 1983.
3. I. Adiri and B. Avi-Itzhak. A time-sharing queue with a finite number of customers. *Journal of the Association for Computing Machinery*, 16(2):315-323, April 1969.
4. Hirotugu Akaike. Markovian representation of stochastic processes and its application to the analysis of autoregressive moving average processes. *Annals of the Institute of Statistical Mathematics*, 26(3):363-387, 1974.
5. Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716-723, December 1974.
6. Hirotugu Akaike. Stochastic theory of minimal realization. *Transactions on Automatic Control*, AC-19(6):667-674, December 1974.
7. Hirotugu Akaike. Markovian representation of stochastic processes by canonical variables. *SIAM Journal on Control*, 13(1):162-173, January 1975.
8. Hirotugu Akaike. Canonical correlation analysis of time series and the use of an information criterion. In R. K. Mehra and D. J. Lainiotis, Editors, *System Identification: Advances and Case Studies*, pages 27-96. Academic Press, New York, 1976.
9. William N. Anderson et al. Consistent estimates of the parameters of a linear system. *The Annals of Mathematical Statistics*, 40(6), December 1969.
10. Jerry Banks and John S. Carson II. *Discrete-Event System Simulation*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1984.
11. George E. P. Box and Gwilym M. Jenkins. *Time Series Analysis, Forecasting, and Control*. Holden-Day Publishers, Oakland, CA, 1976.
12. John M. Charnes. *Statistical Analysis of Multivariate Discrete-Event Simulation Output*. PhD thesis, University of Minnesota, Minneapolis, MN, 1989.
13. Bor-Chung Chen and Robert G. Sargent. Confidence interval estimation for the parameter of stationary processes. *Management Science*, 36(2):200-211, February 1990.
14. R. C. H. Cheng. A note of the effect of initial conditions on a simulation run. *Operations Research Quarterly*, 27(2):467-470, 1976.
15. R. W. Conway. Some tactical problems in digital simulation. *Management Science*, 10(1):47-61, October 1963.

16. David M. Cooper and Eric F. Wood. Identification of multivariate time series and multivariate input-output models. *Water Resources Research*, 18(4):937-946, August 1982.
17. David M. Cooper and Eric F. Wood. Identifying multivariate time series models. *Journal of Time Series Analysis*, 3(3):153-164, 1982.
18. William R. Dillon and Matthew Goldstein. *Multivariate Analysis: Methods and Applications*. John Wiley and Sons, New York, 1984.
19. Samuel Eilon and I. G. Chowdhury. A note on steady-state results in queueing and job-shop scheduling. *Simulation*, 23(3):85-87, September 1974.
20. George S. Fishman. Estimating sample size in computing simulation experiments. *Management Science*, 18(1):21-38, September 1971.
21. George S. Fishman. Bias considerations in simulation experiments. *Operations Research*, 20(4):785-789, July-August 1972.
22. George S. Fishman. Grouping observations in digital simulation. *Management Science*, 24(5):510-521, January 1978.
23. George S. Fishman. *Principles of Discrete Event Simulation*. John Wiley & Sons, New York, 1978.
24. A. V. Gafarian et al. Evaluation of commonly used rules for detecting 'steady state' in computer simulation. *Naval Research Logistics Quarterly*, 25:511-529, 1978.
25. Mark A. Gallagher and Albert H. Moore. Robust minimum-distance estimation using the 3-parameter Weibull distribution. *IEEE Transactions on Reliability*, 39(5):575-580, 1990.
26. F. F. Gan. An optimal design of cusum quality control charts. *Journal of Quality Technology*, 23(4):279-286, 1991.
27. Robert L. Goodrich and Eric A. Stellwagen. *Forecast Master: Multivariate Time Series Forecasting*. Scientific Systems, Cambridge, Massachusetts, 1986.
28. Paul E. Green. *Analyzing Multivariate Data*. The Dryden Press, Hinsdale, Illinois, 1978.
29. Peter Grier. Improving systems with simulators. *Air Force Magazine*, 73(8):40-43, August 1990.
30. J. M. Harrison and R. J. Williams. Brownian models of open queueing networks with homogeneous customer populations. *Stochastics*, 22:77-115, 1987.
31. P. J. Harrison and C. F. Stevens. A bayesian approach to short-term forecasting. *Operations Research Quarterly*, 22:341-362, 1971.
32. P. J. Harrison and C. F. Stevens. Bayesian forecasting. *Journal of the Royal Statistical Society*, 38:205-247, 1976. Serial B.

33. Andrew C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, New York, 1989.
34. Karl P. Hentz. Feasibility analysis of moving bank multiple adaptive estimation and control algorithms. Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1984.
35. Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research*. McGraw-Hill Publishing Company, New York, 1990.
36. William W. Hines and Douglas C. Montgomery. *Probability and Statistics in Engineering and Management Science*. John Wiley & Sons, Inc., New York, NY, 1980.
37. Randall B. Howard. Confidence interval estimation for the output of discrete-event simulations using the Kalman filter. Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, March 1992.
38. Randall B. Howard et al. Confidence interval estimation for univariate output of discrete-event simulations using the Kalman filter. Working paper, Department of Operational Sciences, Air Force Institute of Technology, Wright-Paterson AFB, OH, 1992.
39. Randall B. Howard et al. Confidence intervals for univariate discrete-event simulation output using the Kalman filter. In *Proceedings of the 1992 Winter Simulation Conference*, Washington, DC, 1992. Institute of Electrical and Electronics Engineers.
40. R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:34-45, 1960.
41. R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83:95-108, 1961.
42. W. David Kelton. Transient exponential-Erlang queues and steady-state simulation. *Communications of the ACM*, 28(7):741-749, July 1985.
43. W. David Kelton. Random initialization methods in simulation. *IIE Transactions*, 21(4):355-367, December 1989.
44. W. David Kelton and Averill M. Law. A new approach for dealing with the startup problem in discrete event simulation. *Naval Research Logistics Quarterly*, 30:641-658, 1983.
45. W. David Kelton and Averill M. Law. An analytical evaluation of alternative strategies in steady-state simulation. *Operations Research*, 32(1):169-184, January-February 1984.
46. W. David Kelton and Averill M. Law. The transient behavior of the $M/M/S$ queue with implications for steady-state simulation. *Operations Research*, 33(2):378-396, March-April 1985.
47. Jack P. C. Kleijnen. *Statistical Tools for Simulation Practitioners*. Marcel Dekker, Inc., New York, 1987.

48. Demetrios G. Lainiotis. Optimal adaptive estimation: Structure and parameter adaptation. *IEEE Transactions on Automatic control*, AC-16(2):160-170, April 1971.
49. Demetrios G. Lainiotis. Partitioning: A unifying framework for adaptive systems, I: Estimation. *Proceedings of the IEEE*, 64(8):1126-1143, August 1976.
50. Averill M. Law. Statistical analysis of simulation output data. *Operations Research*, 31(6):983-1029, November 1983.
51. Averill M. Law and John S. Carson. A sequential procedure for determining the length of a steady-state simulation. *Operations Research*, 27(5):1011-1025, September-October 1979.
52. Averill M. Law and W. David Kelton. Confidence intervals for steady-state simulation: I. *Operations Research*, 32(6):1221-1239, November 1984.
53. Averill M. Law and W. David Kelton. *Simulation Modeling and Analysis*. McGraw-Hill Inc., New York, 1991.
54. Robert C. K. Lee. *Optimal Estimation, Identification, and Control*. The M.I.T. Press, Cambridge, Massachusetts, 1964.
55. James M. Lucas. The design and use of V-mask control schemes. *Journal of Quality Technology*, 8(1):1-12, 1976.
56. Albert Madansky. Optimal initial conditions for a simulation problem. *Operations Research*, 24(3):572-577, May-June 1976.
57. Peter S. Maybeck. *Stochastic Models, Estimation, and Control Volume 1*. Academic Press, Inc., New York, 1979.
58. Peter S. Maybeck. *Stochastic Models, Estimation, and Control Volume 2*. Academic Press, Inc., New York, 1982.
59. Peter S. Maybeck. *Stochastic Models, Estimation, and Control Volume 3*. Academic Press, Inc., New York, 1982.
60. Peter S. Maybeck. Adaptive tracking of dynamic airborne vehicles based on (FLIR) image plane intensity data. In *Proceedings of the Third Bad Honnef Conference on Stochastic Differential Systems*, pages 284-305, Berlin, 1986. Springer-Verlag.
61. Peter S. Maybeck and Karl P. Hentz. Investigation of moving-bank multiple model adaptive algorithms. *AIAA Journal of Guidance, Control, and Dynamics*, 10(1):90-96, January-February 1987.
62. Peter S. Maybeck and Michael R. Schore. Robustness of a moving-bank multiple model adaptive algorithm for control of a flexible spacestructure. In *Proceedings of the IEEE National Aerospace and Electronics Conference*, pages 368-374, Dayton, Ohio, 1990.
63. Peter S. Maybeck and Richard D. Stevens. Reconfigurable flight control via multiple model adaptive control methods. *IEEE Transactions on Aerospace and Electronic Systems*, AES-27(3):470-480, 1991.

64. R. K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7:637-640, 1971.
65. Raman K. Mehra. On the identification of variances and adaptive Kalman filtering. *IEEE Transactions of Automatic Control*, AC-15(2):175-184, April 1970.
66. Raman K. Mehra. On-line identification of linear dynamic systems with applications to Kalman filtering. *IEEE Transactions of Automatic Control*, AC-16(1):12-21, February 1971.
67. R. J. Meinhold and N. D. Singpurwalla. Robustification of Kalman filter models. *Journal of the American Statistician*, 84:479-486, June 1989.
68. Richard. J. Meinhold and Nozer. D. Singpurwalla. Understanding the Kalman filter. *Journal of the American Statistician*, 37(3):123-127, May 1983.
69. Marc S. Meketon and Bruce Schmeiser. Overlapping batch means: Something for nothing? In *Proceedings of the 1984 Winter Simulation Conference*, pages 227-230, Atlanta, GA, 1984. Institute of Electrical and Electronics Engineers.
70. Douglas C. Montgomery. *Introduction to Statistical Quality Control, 2nd Edition*. John Wiley & Sons, Inc., New York, NY, 1991.
71. Joseph R. Murray and W. David Kelton. The transient behavior of the $M/E_k/2$ queue and steady-state simulation. *Computers and Operations Research*, 15(4):357-367, 1988.
72. Salih N. Neftci. Specification of economic time series models using Akaike's criterion. *Journal of the American Statistical Association*, 77(379):537-540, September 1982.
73. Amedeo R. Odoni and Emily Roth. An empirical investigation of the transient behavior of stationary queueing systems. *Operations Research*, 31(3):432-455, May-June 1983.
74. Department of Defense. *Critical Technologies Plan*. Committees on Armed Services. United States Congress, 1990.
75. Emanuel Parzen. *Stochastic Processes*. Holden-Day, Inc., San Francisco, 1962.
76. Charles H. Porter. Comparison of batch means and independent replications techniques to applications of the Kalman filter for simulation output analysis. Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, March 1990.
77. A. Alan. B. Pritsker. *Introduction to Simulation and SLAM II*. Systems Publishing Corporation, West Lafayette, IN, 1986.
78. Robert Roehrkasse and George C. Huges. Crisses analysis: Operation Desert Shield. *OR/MS Today*, 17(6):22-27, December 1990.
79. Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, Inc., New York, 1989.
80. R. Y. Rubinstein and R. Marcus. Efficiency of multivariate control variates in monte carlo simulation. *Operations Research*, 33(3):661-677, May-June 1985.

81. Robert G. Sargent and Tapas K. Som. Current issues in frequency domain experimentation. *Management Science*, 38(4):667-687, 1992.
82. B. Schmeiser and W. T. Song. Correlation among estimators of the variance of the mean. In *Proceedings of the 1987 Winter Simulation Conference*, pages 309-317, Atlanta, GA, 1987. Institute of Electrical and Electronics Engineers.
83. Bruce Schmeiser. Batch size effects in the analysis of simulation output. *Operations Research*, 30(3):556-568, May-June 1982.
84. Thomas Schriber. *Simulation Using GPSS*. Wiley Publishing Company, New York, 1974.
85. Thomas J. Schriber and Richard W. Andrews. ARMA-based confidence intervals for simulation output analysis. *American Journal of Mathematical and Management Sciences*, 4(3-4):345-373, 1984.
86. Lee W. Schruben. Control of initialization bias in multivariate simulation response. *Communications of the ACM*, 24(4):246-252, April 1981.
87. Lee W. Schruben. Detecting initialization bias in simulation output. *Operations Research*, 30(3):569-590, May-June 1982.
88. Lee W. Schruben et al. Optimal tests for initialization bias in simulation output. *Operations Research*, 31(6):1167-1177, November-December 1983.
89. Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461-464, 1978.
90. Andrew F. Seila. Multivariate simulation output analysis. *American Journal of Mathematical and Management Sciences*, 4(3 & 4):313-334, 1984.
91. Andrew F. Seila. Multivariate estimation of conditional performance measures in regenerative simulation. *American Journal of Mathematical and Management Sciences*, 10(1 & 2):17-50, 1990.
92. Robert E. Shannon. Simulation: A survey with research suggestions. *AIIE Transactions*, pages 289-301, September 1975.
93. Robert E. Shannon. *Systems Simulation: The Art and Science*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1975.
94. Stuart N. Sheldon and Peter S. Maybeck. An optimizing design strategy for multiple model adaptive estimation and control. In *SIAM Conference on Control in the 90's: Achievements, Opportunities, and Challenges*, San Francisco, CA, 1989.
95. Donald L. Snyder. *Random Point Processes*. Wiley Publishing Company, New York, 1975.
96. M. A. Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730-737, 1974.

Vita

Captain Mark Austin Gallagher was born on August, 18, 1961 in Saint Paul, Minnesota. He graduated from the United States Air Force Academy in 1983 with a Bachelor of Science in Operations Research and Computer Science. As a lieutenant, he served as a cost analyst in Aeronautical System Division in Dayton, Ohio. He received his Master's of Science in Operations Research from the Air Force Institute of Technology (AFIT) in 1986. Afterward, he was assigned to the Targeting Directorate of Headquarters, Strategic Air Command. In 1989, he returned to AFIT to begin his doctoral studies. After completing his Ph.D., Captain Gallagher was assigned to the Weapon System Cost Division of the Under Secretary of Defense for Program Analysis and Evaluation. He is married to the former Miss Nancy Victoria Austin.

Permanent address: 2110 South Fillmore Street
Arlington, Virginia 22204

97. Harold J. Steudel and S. M. Wu. A time series approach to queueing systems with applications for modeling job-shop in-process inventories. *Management Science*, 23(4):745-755, 1977.
98. Ruey S. Tsay. Identifying multivariate time series models. *Journal of Time Series Analysis*, 10(4):357-372, 1989.
99. Ruey S. Tsay and George C. Tiao. Use of canonical analysis in time series model identification. *Biometrika*, 72(2):299-315, 1985.
100. Mark A. Turnquist and Joseph M. Sussman. Toward guidelines for designing experiments in queueing simulation. *Simulation*, 28:137-144, May 1977.
101. Peter D. Welch. The statistical analysis of simulation results. In S. S. Laverberg, Editor, *The Computer Performance Modeling Handbook*, pages 268-328. Academic Press, New York, 1983.
102. Ward Whitt. Planning queueing simulations. *Management Science*, 35(11):1341-1366, November 1989.
103. James R. Wilson. Statistical aspects of simulation. In J. P. Brans, Editor, *Operational Research '84: Proceedings of the Tenth International Conference on Operational Research*, pages 921-937, Amsterdam, 1984. Elsevier Science Publishers B.V. (North Holland).
104. James R. Wilson and A. Alan B. Pritsker. Evaluation of startup policies in simulation experiments. *Simulation*, 31(3):79-89, September 1978.
105. James R. Wilson and A. Alan B. Pritsker. A survey of research on the simulation startup problem. *Simulation*, 31(2):55-58, August 1978.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1992		3. REPORT TYPE AND DATES COVERED Doctoral Dissertation
4. TITLE AND SUBTITLE Identification of the Initial Transient in Discrete-Event Simulation Output Using the Kalman Filter			5. FUNDING NUMBERS	
6. AUTHOR(S) Mark A. Gallagher, Captain, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/DS/ENS/92-1	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Dr. Neal Glassman AFOSR/NM Bolling AFB, DC 20332-6448			10. SPONSORING / MONITORING AGENCY REPORT NUMBER 90NM429	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Data truncation is a commonly accepted method of dealing with initialization bias in discrete-event simulation. Algorithms for determining the appropriate initial-data truncation point for univariate and multivariate output are proposed. The techniques entail averaging across replications and estimating a steady-state output model in a state-space framework. Using the estimated model, Multiple Model Adaptive Estimation (MMAE), which uses Kalman filters with different parameter vectors, is applied. Based on the filters' residuals, the conditional probabilities of each filter's specific parameter vector being correct are determined. The MMAE parameter estimates are the probabilistic-weighted average of the filters' assumed parameter vectors. The estimated truncation point is selected when a vector norm of the difference between an estimate of the steady-state mean vector and the MMAE time-varying estimated mean vector are within a small tolerance. A Monte Carlo analysis using data generated from discrete-event simulations are used to evaluate the techniques. The evaluation criteria include the ability to accurately estimate and to construct reliable confidence regions for the vector of the response means based on the truncated sequences.				
14. SUBJECT TERMS Simulation, bias, truncation, Kalman filter, Multiple Model Adaptive Estimation			15. NUMBER OF PAGES 297	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	